

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Návrh a vývoj systému pro analýzu kognitivních schopností

Design and Development of Systems for the Analysis of Cognitive Abilities

Zadání bakalářské práce

Student: **Jakub Sekanina**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Návrh a vývoj systému na analýzu kognitivních schopností**
Design and Development of Systems for the Analysis of Cognitive Abilities

Jazyk vypracování: čeština

Zásady pro vypracování:

Vývoj SW platformy na testování kognitivních funkcí na základě modelu neuropsychologického rozdělení kognitivních funkcí (Range of Abilities Evaluated by Neuropsychological Assessment). Za testované neuropsychologické kognitivní funkce byly vybrány tyto oblasti a procesy: vnímání (percepce), paměti (mnestické funkce, zahrnující procesy učení), zrakově prostorové schopnosti (visuospeciální), exekutivní funkce (zahrnující schopnosti plánování, rozhodování, vůle, seberegulace, propojení podnětů, myšlení aj.) a rychlosti zpracování.

Cíle práce:

1. Nastudování webových aplikací, ochrany a archivaci patientských dat (anonymizace dat).
2. Nastudování problematiky testování kognitivních funkcí.
3. Problematika diagnostiky kognitivních funkcí přes PC rozhraní, výstupy do neurorehabilitačních programů.
4. Realizace webového portálu obsahující diagnostický nástroj vybraných kognitivních funkcí, a jeden užívající dotazník (CFQ) pokrývající problematiku kognitivních chyb.
5. Ověření funkčnosti na 30 zdravých probandech včetně základního statistického zpracování měřených dat a zhodnocení souvztažnosti s užitým dotazníkem.
6. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] Plháková, A. Učebnice obecná psychologie. Praha: Academia, 2007.
- [2] URBÁNEK, Tomáš, Denisa DENGLEROVÁ a Jan ŠIRŮČEK. Psychometrika: měření v psychologii. Praha: Portál, 2011, 319 s. ISBN 978-80-7367-836-4.
- [3] Vágnerová M. Obecná psychologie. Praha: Karolinum, 2016.
- [4] SMOLINSKI, Tomasz G., Mariofanna G. MILANOVA a Aboul Ella. HASSANIEN. Computational intelligence in biomedicine and bioinformatics: current trends and applications. New York: Springer, c2008. Studies in computational intelligence, v. 151. ISBN 978-3-540-70778-3.
- [5] CAZALS, Frederic. a Pierre. KORNPROBST. Modeling in computational biology and biomedicine: a multidisciplinary endeavor. New York: Springer, 2013. ISBN 978-3-642-31208-3.
- [6] SURESH R. DEVASAHAYAM. Signals and systems in biomedical engineering signal processing and physiological systems modeling. 2nd ed. New York: Springer, 2013. ISBN 9781461453321.
- [7] DEVASAHAYAM, Suresh R. Signals and systems in biomedical engineering: signal processing and physiological systems modeling. Second edition. New York: Springer, 2013. ISBN 978-1-4614-5331-4.
- [8] DESERNO, Thomas M. Biomedical image processing. Heidelberg: Springer, c2011. Biological and

medical physics, biomedical engineering. ISBN 978-3-642-15816-2.

[9]DOUGHERTY, Geoff. Medical image processing: techniques and applications. New York: Springer, c2011. Biological and medical physics, biomedical engineering. ISBN 978-1-4419-9779-1.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **prof. Ing. Lačezar Ličev, CSc., prof.h.c.**

Konzultant bakalářské práce: Ing. Martin Augustynek, Ph.D.

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2019



doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry





prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2019

.....*Salm*.....

Rád bych poděkoval prof. Ing. Lačezaru Ličevovi, CSc., prof.h.c. a doc. Ing. Martinu Augustynkovi, Ph.D., za odbornou pomoc a trpělivost při konzultacích, PhDr. Petru Niliusovi, Ph.D., za pomoc při tvorbě testu a vyhodnocení výsledků a také všem dalším lidem, kteří mi poskytli cenné rady k tomu, abych tuto práci zvládl dotáhnout až do konce.

Abstrakt

Předmětem této bakalářské práce je návrh a vývoj webové aplikace sloužící jako diagnostický nástroj kognitivních funkcí s předpokladem využití pro pacienty se získaným poškozením mozku. Práce je rozdělena na teoretickou a praktickou část. V rámci teoretické části je zpracována problematika vybraných oblastí kognitivních funkcí a jejich testování v psychologii včetně využití počítačových technologií. Dále je v této části rozebírána problematika webových aplikací a možnosti ochrany a archivace uživatelských dat. Předmětem praktické části je samotný návrh systému — ASP.NET webové aplikace a diagnostického nástroje zaměřujícího se na testování vizuálního zpracovávání (vnímání), fungování pracovní paměti ve vizuální paměti, pozdějšího vybavení naučeného, exekutivních funkcí (schopnosti plánování a udržení niterného plánu) a celkového psychomotorického tempa. Vytvořený test byl porovnán s dotazníkem kognitivních chyb (CFQ). Výsledkem je klasifikace výkonu probanda a klasifikace výsledků dle uspořádání CHC modelu inteligence. Aplikace byla testována a statisticky ověřována na 106 zdravých probandech formou online administrace paralelních verzí a v rámci experimentálního užití EEG brain mappingu na zdravém subjektu.

Klíčová slova: aplikace na analýzu kognitivních schopností, webové aplikace, testování v psychologii, ochrana dat

Abstract

The subject of this thesis is design and development of a web application serving as a diagnostic tool of cognitive functions for patients with acquired brain damage. The thesis is divided into a theoretical and practical part. The theoretical part is focused on the issue of selected cognitive functions and their testing in psychology including the use of computer technology. Following is the issue of web applications and data security. Subject of the practical part is design of the system itself — an ASP.NET web application and diagnostic tool capable of testing visual processing and perception, working memory, long term memory and retrieval, executive functions (the ability to plan and maintain intimate plan) and overall psychomotor speed. The resulting cognitive test was associated with standardized Cognitive Failures Questionnaire (CFQ) and administered to 106 healthy probands in an online setting with the use of parallel versions. The results were classified according to CHC model of intelligence. The cognitive test was also administered in context of EEG brain mapping.

Key Words: application for cognitive functions diagnostics, web applications, testing in psychology, data security

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	11
Seznam tabulek	12
Seznam výpisů zdrojového kódu	13
1 Úvod	14
2 Kognitivní funkce	15
2.1 Vnímání	15
2.2 Paměť	16
2.3 Myšlení	16
2.4 Rychlost zpracování informací	17
3 Testování kognitivních funkcí	18
3.1 Psychologický test	18
3.2 Historie testování	19
3.3 Chyby v měření při testování	21
4 Testování kognitivních funkcí s využitím počítačů	24
4.1 Typy využití	24
4.2 Výhody a nevýhody	25
4.3 Příklady aplikací měřících kognitivní funkce	25
4.4 Výstupy do neurorehabilitačních aplikací	26
5 Webové aplikace	28
5.1 HTTP protokol	28
5.2 Technologie webových aplikací na straně klienta	29
5.3 Technologie webových aplikací na straně serveru	31
6 Ochrana a archivace dat	34
6.1 GDPR	34
6.2 Možnosti ochrany dat ve webových aplikacích	35
7 Realizace webového portálu obsahujícího nástroj pro měření kognitivních funkcí	41
7.1 Specifikace požadavků	41
7.2 Návrh systému – webový portál	42

7.3	Návrh systému – diagnostický nástroj	48
8	Ověření funkčnosti navrženého řešení	59
8.1	Ověření funkčnosti	59
8.2	Statistické zpracování měřených dat	60
8.3	Zhodnocení souvztažnosti dotazníku CFQ s testem	64
8.4	Vyhodnocení měřených dat ve webové aplikaci	65
9	Závěr	66
	Literatura	68
	Přílohy	71
A	Ukázka vyhodnocení testu	72
B	Grafy korelací dvou měřených oblastí CHC modelu	73
C	Grafy zobrazené ve webové aplikaci	74
D	Seznam elektronických příloh	75

Seznam použitých zkratek a symbolů

AI	– Artificial Intelligence
AJAX	– Asynchronous JavaScript and XML
API	– Application Programming Interface
ASP	– Active Server Pages
CFQ	– Cognitive Failures Questionnaire (dotazník kognitivních chyb)
CHC	– Cattell–Horn–Carroll theory
CLR	– Common Language Runtime
CMS	– Content Management System
CSRF	– Cross-site Request Forgery
CSS	– Cascading Style Sheets
DLL	– Dynamic-link library
DOM	– Document Object Model
EEG	– electroencephalogram
EU	– Evropská unie
GC	– Garbage Collection
GDPR	– General Data Protection Regulation
GWT	– Google Web Toolkit
HDD	– Hard Disk Drive
HTML	– HyperText Markup Language
HTTP	– HyperText Transfer Protocol
HTTPS	– HyperText Transfer Protocol Secure
I/O	– Input/Output (vstup/výstup)
ID	– Identifier
IIS	– Internet Information Services
IP	– Internet Protocol
IQ	– Intelligence Quotient
JVM	– Java Virtual Machine
JSON	– JavaScript Object Notation
JSF	– JavaServer Faces
JSP	– JavaServer Pages
LAMP	– Sada softwaru – Linux, Apache, MySQL, PHP
LESS	– Leaner Style Sheets
MAC	– Media Access Control
MD5	– Message-digest algorithm verze 5
MVC	– Model-View-Controller
NSA	– National Security Agency

PHP	– PHP: Hypertext Preprocessor, drive Personal Home Page
RAID	– Redundant Array of Independent Disks
SASS	– Syntactically Awesome Style Sheets
SHA	– Secure Hash Algorithm
SPA	– Single Page Application
SQL	– Structured Query Language
SQLi	– SQL Injection
SVG	– Scalable Vector Graphics
T-SQL	– Transact-SQL
URL	– Uniform Resource Locator
VPN	– Virtual Private Network
XSS	– Cross-site Scripting
npm	– Node.js Package Manager

Seznam obrázků

1	Zákony organizace	15
2	Rorschachův test inkoustových skvrn [8]	20
3	NELA – Bourdonův test	26
4	Asymetrické šifrování	36
5	Use case diagram webového portálu	43
6	Registrační formulář	44
7	Postup registrace uživatele do systému	46
8	Uživatelské rozhraní zálohy a obnovení databáze	47
9	Jednotná tlačítka napříč celým webem	47
10	Princip průběžného přeposílání výsledků na server	49
11	Ukázka subtestu vnímání	51
12	Ukázka subtestu pamětního procesu vštípení	52
13	Ukázka subtestu zaměřeného na proces plánování	53
14	Grafické zobrazení mozkové aktivity člověka při vykonávání subtestu s Hanoj- skými věžemi [46]	59
15	Graf distribuce počtu správně uhodnutých figur	61
16	Graf distribuce počtu bodů	62
17	Graf distribuce celkového počtu tahů	63
18	Krabicový graf času prvního kola	64
19	Korelace CFQ dotazníku a testu	65
20	Vyhodnocení testu	72
21	Graf korelace kvantitativních schopností	73
22	Graf korelace zrakových procesů	73
23	Sloupcový graf distribuce měřených položek v aplikaci	74
24	Histogram distribuce měřených položek v aplikaci	74

Seznam tabulek

1	Statistické veličiny vypočtené z počtu správně uhodnutých figur	61
2	Statistické veličiny vypočtené z počtu bodů ze subtestu pamětního procesu vštípení	62
3	Statistické veličiny vypočtené z celkového počtu tahů	63
4	Statistické veličiny vypočtené z času prvního kola (v sekundách)	64

Seznam výpisů zdrojového kódu

1	HTTP požadavek na hlavní stránku webu seznam.cz	29
2	HTTP odpověď z webu seznam.cz	29
3	Ukázka HTML elementu img	30
4	CSS selektor a tělo s vlastnostmi	31
5	C# kód pro formátování SQL dotazu	38
6	JavaScriptová funkce upravující CSS vlastnost blur	51
7	JavaScriptová funkce na zobrazení motýla na náhodné pozici	52
8	JavaScriptová funkce na aktualizování grafického výstupu Hanojských věží	53
9	Minimax s alfa-beta ořezáváním implementovaný v JavaScriptové funkci	55

1 Úvod

Lidská psychika má jednu vlastnost — velice špatně se měří její výkon. Pokud bychom chtěli změřit, jak velkou má konkrétní člověk paměť, či jaká je jeho rychlost reakcí, jaký nástroj bychom k tomu použili?

Bohužel v dnešní době neexistuje možnost člověku doslova přečíst mysl a tak se můžeme spoléhat pouze na nepřímé měření lidské mysli. Tato bakalářská práce se zabývá právě tímto nepřímým měřením lidských kognitivních funkcí prostřednictvím počítačového rozhraní. Cílem tohoto nástroje, který se o měření bude starat, je analyzovat konkrétní množinu kognitivních funkcí člověka a následně mu říci, jak si na tom stojí v rámci vzorku lidí, kteří tento test taktéž provedli.

Analýza bude probíhat skrze dotazník kognitivních chyb a následně skrze test zaměřený na konkrétní kognitivní funkce. Celý systém poběží uvnitř webové aplikace, která bude sloužit jako rozhraní pro uživatele, kteří tento test vykonají. Bude zde mimo jiné kladen důraz i na ochranu patientských dat.

Cílem předkládané práce je, aby se tento nástroj dal v budoucnu napojit jako modul do neurorehabilitačního softwaru Eddie, kde bude sloužit jako vstupní test pro pacienty s poškozením mozku. Na základě výsledku testu pak bude možné pacientovi přiřadit adekvátní úlohy na procvičování problémových kognitivních funkcí.

Bakalářská práce se dělí na dvě části. První kapitoly se zabývají kognitivními funkcemi a jejich testováním v psychologii a následně je popsána důležitá teorie webových aplikací a ochrany uživatelských dat.

Druhá část práce se zabývá především návrhem webové aplikace a její implementací včetně realizace a implementace diagnostického nástroje pro hodnocení kognitivních funkcí.

V závěru předkládané práce je zhodnocen výsledek nasazení aplikace a jsou diskutovány výsledky testu, který byl ověřen na zdravých probandech. Závěrečné zhodnocení obsahuje také hodnocení souvztažnosti dotazníku CFQ s testem a způsob, jakým byla v samotném webovém portálu implementována analýza testovaných dat.

2 Kognitivní funkce

Jeden z klíčových procesů lidského psychického procesu jsou kognitivní funkce. Úlohou kognitivních funkcí je zpracovávání, ukládání, transformace a užití informací přicházejících z vnějšího světa. Mezi tyto kognitivní funkce se řadí velké množství procesů, které tyto operace umožňují. [1]

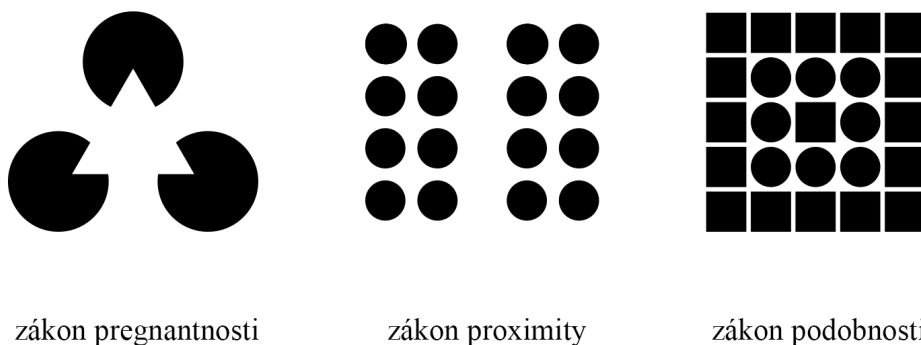
Dělení kognitivních funkcí není však jednoznačně určené, v tomto textu se zaměřuji na vnímání, paměť, myšlení a rychlost zpracování informací. Tyto kognitivní funkce hrají velkou roli v diagnostickém nástroji, který je implementován v rámci této práce.

2.1 Vnímání

Naše smysly zaznamenávají různé druhy informací z okolního světa. Viditelné světlo o frekvencích, které jsou odráženy konkrétními předměty, dopadá do lidského oka, kde je následná světelná informace převedena na nervový vzruch. Rozvibrované částice kolidují s bubínkem v lidském uchu, který se následně rozvibruje ekvivalentní frekvencí a dalším procesem se tato vibrace převede na elektrický signál vedený nervovou soustavou.

Tyto a další informace z vnějšího světa zachytávané našimi smyslovými orgány však musí první být určitým způsobem organizovány a interpretovány, aby nám dávaly nějaký smysl. Přesně tohle má na starost kognitivní proces zvaný vnímání. [2]

Existují dvě hlavní teorie vnímání. Jedna považuje vnímání za komplikovanější psychický děj, při kterém se využívá nabitých zkušeností skrze učení a výstupů z jiných kognitivních procesů. Druhá teorie naopak na vnímání pohlíží jako na přímý proces. [2]



Obrázek 1: Zákony organizace

V Německu se vnímáním detailně zabývala tvarová psychologie, tzv. *gestaltismus*. Představitelé tohoto směru došli k zjištění, že lidské vnímání se řídí určitými pravidly. Ať už se jedná o jakýkoliv lidský smysl, informace je vždy nějakým způsobem upravena a organizována. Tyto druhy organizace jsou definovány v tzv. *zákonech organizace*. Mezi ně patří například *zákon*

pregnantnosti (neúplné tvary máme tendenci vnímat jako úplné), *zákon proximity* (prvky, které jsou blíže u sebe, vnímáme jakoby patřily k sobě) nebo třeba *zákon podobnosti* (věci, které jsou si podobné, vnímáme jako celky). [2]

2.2 Paměť

Paměť se stará o zaznamenávání zažitých životních okamžiků. Jako všechny ostatní kognitivní procesy, je i paměť nepostradatelnou funkcí lidské psychiky. Samotný proces paměti je možné rozdělit na tři základní fáze: [2]

- **Vštípení** je proces, kdy se konkrétní informace zakóduje a zařadí do lidské psychiky tak, že ji bude možné později znovu v této pomyslné knihovně informací najít.
- **Retence** je aktivním procesem udržování této informace. V průběhu tohoto procesu dochází k neustálému zpracovávání a třídění uchovávaných informací.
- **Reprodukce** nebo-li vybavení je proces, ke kterému nastává vždy, když konkrétní uchovávanou informaci v paměti potřebujeme použít.

Paměť se také dělí na několik složek: [2]

- **Senzorická paměť** je složka paměti, která uchovává informace na nejkratší možnou dobu, která se pohybuje v jednotkách sekund. Důležité informace, které se dostanou do senzorické paměti, jsou následně přesunovány do krátkodobé paměti, kde je s nimi možné dále pracovat.
- **Krátkodobá paměť** slouží jako pracovní paměť. Uchovává informace, se kterými v daný moment pracujeme. Kapacita této složky paměti je okolo sedmi položek a délka zapamatování se pohybuje okolo patnácti až třiceti sekund.
- **Dlouhodobá paměť** se stará o dlouhodobé uložení informací. Uložení informací do dlouhodobé paměti je ovlivněno mnoha faktory. Je známo, že v případě silných emotivních zážitků může dojít k narušení procesu ukládání a tedy k zneřecnění informace či k jejímu úplnému vyřazení z paměti.

2.3 Myšlení

Tento kognitivní proces představuje obzvláště práci s informacemi. Cílem myšlení je dojít s pomocí dostupných informací k novému poznatku. Inteligence popisuje kvalitu myšlení a úroveň abstrakce, v jaké myšlení u daného jedince probíhá. Myšlení se dá rozdělit na tři typy: [2]

- **Konkrétní myšlení** užívané při zpracovávání vjemů. Jedná se o operace s konkrétními předměty ve vnějším světě. Příkladem může být vaření nebo třeba řízení auta.

- **Názorné myšlení** využívá místo konkrétních předmětů našich představ. Příkladem názorného myšlení je hra piškvorky, kdy je třeba naplánovat náš další tah tak, abychom nedali protivníkovi šanci zvítězit.
- **Abstraktní myšlení** je využito při práci se zástupnými symboly — abstraktními pojmy. Snažit se pochopit matematickou definici je příkladem abstraktního myšlení.

Mezi lidmi dochází i jako u jiných živých bytostí ke komunikaci. Bez komunikace bychom nebyli schopni varovat své druhy o blížícím se nebezpečí, neexistovala by spolupráce, nemohli bychom sdílet své myšlenky s ostatními. Na rozdíl od jiných živočichů si však lidé vyvinuli mnohem důmyslnější jazyk, který odráží komplexita lidských bytostí. Jazyk nám umožňuje předat výsledky či samotný proces myšlení druhým lidem a společně tento proces případně i urychlit.

2.4 Rychlost zpracování informací

Rychlost zpracování informací je dle některých zdrojů označována jako kognitivní schopnost [3] a dle jiných jako vlastnost, která popisuje rychlost reakce jedince na konkrétní podnět. [4] Rychlost zpracování informací protíná různé oblasti lidské psychiky. U lidské paměti se může jednat o rychlost vybavování, u vnímání o rychlost analýzy obrazu vstupujícího přes smysly.[4]

Pomalá rychlost zpracování informací není spojená s inteligencí. Pomalejší člověk může dojít ke stejnému výsledku jako jiní, avšak mu to zabere delší dobu. Stárnutím se rychlost zpracování informací přirozeně snižuje. [3, 4]

3 Testování kognitivních funkcí

Pokud chceme zjistit výšku člověka, jeho váhu, nebo věk, je obecně jasné, jak se k takovým údajům dostat — člověka změříme metrem, zvážíme váhou a věk spočítáme ze znalosti jeho data narození. Co když ale potřebujeme určit, jak je na tom konkrétní osoba třeba s pamětí? Je totiž všeobecnou znalostí, že se lidé navzájem liší, a to nejen vzhledově, tedy zvnějšku, ale i psychickými procesy včetně kognitivního výkonu. Mladší lidé obecně mívají větší kapacitu paměti než lidé vyššího věku, rozdíly jsou však i mezi lidmi podobných věkových kategorií. A pokud existují rozdíly v kapacitě paměti, mělo by být možné tyto rozdíly nějakým způsobem kvantifikovat.

Mimo převod kapacity paměti na číselnou hodnotu lze samozřejmě užít odhadu. Dejme tomu, že jsem schopen odhadnout, jak je na tom má paměť v porovnání s člověkem, kterého velice dobře znám. Tento odhad však může být zkreslený. Zdrojem zkreslení v tomto případě může být fakt, že srovnávám dvě různé situace nebo třeba neberu v potaz okolnosti, které by výkon druhého člověka mohly ovlivnit. Někteří lidé mohou mít mnohem větší problémy vybavit si věci z paměti ve stresujících situacích. Jak pak můžu srovnávat vybavovací schopnosti takového člověka se schopnostmi někoho, kdo je vůči stresu více odolný?

Pokud však navodím ideální podmínky v prostředí, které žádného jedince nebude nijak zvyhodňovat, jsem schopen dosáhnout toho, že zkreslenost takového srovnání bude minimální.

Tyto ideální podmínky jsou utvářeny v rámci psychologického testu.

3.1 Psychologický test

Test, konkrétně psychologický test, má mnoho definic. Jedna z nich definuje test jako měřicí nástroj/techniku používanou ke kvantifikaci lidského chování nebo k pomoci jeho pochopení a predikci. [5] Ostatně test je ten hledaný měřicí nástroj, kterým můžeme měřit, ačkoliv nepřímo, lidské psychické procesy.

Jak ale k takovému měření dochází? Stručně řečeno, psychologický test mívá předem určenou stupnici a jednotlivé položky testu bývají hodnoceny určitým počtem bodů. Po vypočtení celkového počtu bodů daného testu můžeme dojít k číslu — skóre testu.

Skóre jako takové ovšem nic neznamena. Jakou užitečnou informaci získám, když se dozvím, že můj výsledek testu paměti je dvacet sedm bodů z padesáti? Jediná informace, která z tohoto výsledku vyplývá, je mé umístění v rámci hodnotící stupnice, které je v tomto příkladě lehce nad střední hodnotou. Problém je v tom, že si takový výsledek nedokážu s ničím spojit. Znamená dosažených dvacet sedm bodů z padesáti, že je má paměť průměrná?

Věc, která dává výsledku z psychologického testu větší smysl, je srovnání mého skóre s výsledky ostatních lidí, kteří tento test vykonali. [5] Tak jsem pak schopen dojít k zjištění, že průměrné skóre v mém pomyslném testu je třicet bodů a že tedy můj výsledek je pod tímto průměrem.

3.1.1 Dělení psychologických testů

Vzhledem k tomu, že jediný formát testu nedokáže pokrýt všechny případy testování, existuje několik různých podob, do kterých se testy dají dělit.

Jedno z dělení je na testy individuální a skupinové.

- **Individuální test** je zadáván jediné osobě. Testování se podobá psychologickému vyšetření, kdy je předmětem pozorování jedna konkrétní osoba. Tento druh testu umožňuje získání mnohem přesnějších dat, než skupinová obdoba.
- **Skupinový test** je zadáván většímu počtu lidí zároveň. Cílem je ušetřit co nejvíce času a získat mnohem větší, ačkoliv méně přesnější, objem dat. Takové testování se nejvíce podobá klasickým školním testům.

Druhé z možných dělení rozděluje testy na ty, které měří úroveň schopnosti člověka nebo-li výkonnostní testy a na testy, které jsou schopny určit osobnost člověka.

- **Výkonnostní testy** se zaměřují na veličiny jako je rychlost, správnost a přesnost.
- **Osobnostní testy** se snaží určit rysy osobnosti, charakter nebo temperament. Neexistují zde žádné správné či špatné odpovědi. Naopak odpovědi mívají hodnocení dle toho, jak moc jsou relevantní pro daný druh osobnosti.

Každý z těchto typů testů může mít různá užití v praxi. Psychologické testy se provádí při výběru pracovníků, u přijímacích zkoušek na různé vzdělávací instituce nebo třeba při výběru do armády.

Klíčové je ovšem zvláště ve zdravotnictví, kde se uplatní mimo jiné při diagnostice různých psychických poruch či identifikaci problémů u pacientů s poškozením mozku. [5]

3.2 Historie testování

Jedny z prvních zmínek o testování se datují do doby starověké Číny, kdy probíhaly ústní zkoušky pro jedince, kteří stáli o pozici ve veřejné správě. Tento způsob využití testů později inspiroval britskou vládu a poté i vládu ve Francii a Německu, kde se taktéž testováním určovali lidé vhodní na místo ve veřejné správě. [6, 5]

Ve 13. století v Evropě, v době vzniku prvních univerzit, se testování začalo používat k obhajobě titulu a později se začalo rozšiřovat i do zbytku škol. Z ekonomických důvodů zpočátku převládal ústní druh zkoušení, písemně se testovalo až v pozdějších dobách, kdy byl papír mnohem dostupnější. [7]

Jednou z důležitých postav měření a testování je Francis Galton, který byl inspirován teorií evoluce svého bratrance Charlese Darwina. Darwin ve své teorii tvrdil, že živočišné druhy s výhodnějšími znaky mají mnohem větší šanci v přírodě přežít a stihnout přivést na svět své

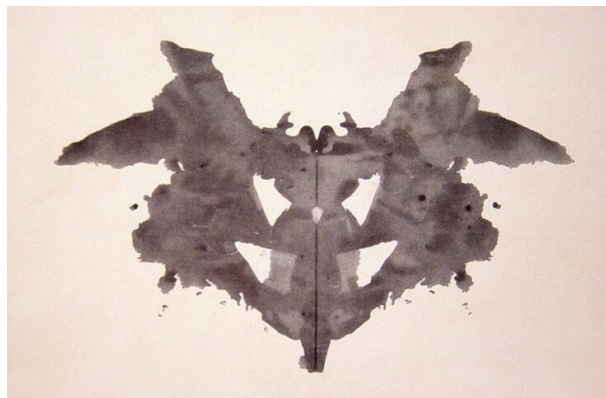
potomky, kteří poté tyto znaky zdědí. Naopak druhy, které takové štěstí neměly, vymřou a s nimi i jejich genetická informace. Galton se pokoušel dokázat, že někteří lidé mají výhodnější znaky, než jiní, zvláště se zaměřoval na inteligenci. Intelligence dle něho je dědičná a je třeba motivovat inteligentní lidi, aby měli potomky. Zvláště se pokoušel navrhnout inteligenční test, který by dokázal inteligentní jedince rozlišit od průměrných. [6, 5, 7]

Na Galtonovy výzkumy inteligence navázal James McKeen Cattell, který se taktéž pokoušel o vytvoření určitého testu inteligence, který svými slovy nazval jako *psychotest* (mental test). Pokusy obou vědců skončily neúspěchem. [7]

První úspěšný test inteligence — Binetovu-Simonovu škálu — vytvořil Alfred Binet spolu s Theodorem Simonem na popud francouzského ministra veřejného vzdělávání, který chtěl rozlišit intelektuálně podprůměrnou populaci, aby jim bylo možné poskytnout adekvátní vzdělání. Výsledkem tohoto testu byl mentální věk, tedy věk, který korespondoval s mentální výkonností testovaného. Vzhledem k velikosti poptávky po testu tohoto druhu byl test překládán do několika jazyků a používán i v jiných zemích. [6, 5, 7]

Velké využití testů si ve Spojených státech vyžádala první světová válka, kdy bylo třeba roztrždit brance do americké armády. Vznikly dva testy, Army Alpha a Army Beta. Druhá verze testu byla určena pro negramotné rekruty. V této době taktéž vznikla Stanfordská Binetova škála, revize Binetovy-Simonovy škály. Výsledkem tohoto testu bylo jediné skóre a to IQ, neboli inteligenční kvocient. [6, 5]

Mezi světovými válkami se staly velmi populární osobnostní testy, zároveň se však začaly objevovat i tzv. projektivní osobnostní testy z důvodu, že čistě osobnostní testy spoléhaly na pravdomluvnost testovaného člověka a to se ukázalo jako problém, který snižoval validitu těchto testů.



Obrázek 2: Rorschachův test inkoustových skvrn [8]

Projektivní osobnostní testy naproti tomu využívaly nejednoznačných podnětů. Jedním z takových testů je např. Rorschachův test inkoustových skvrn, při kterém je testovanému člověku postupně předloženo deset karet se symetrickými nejednoznačnými tvary, vytvořenými přeložením skvrn od inkoustu. Testovaný člověk má pak za úkol říct první věc, která ho napadne, když

se na skvrnu podívá. Proto u projektivních osobnostních testů nehrozilo, že by si pacient odpověď vymyslel. U Rorschachového testu je však často vytýkána jeho nízká validita a reliabilita. [6, 5]

Kurt Goldstein, německý neurolog, prováděl pozorování vojáků, kteří v bojích za první světové války přišli k poškození mozku. Na základě jeho výzkumů se v oboru neuropsychologie v průběhu dalších let objevila řada nástrojů sloužících pro detekci druhu poškození mozku. Mnoho z těchto nástrojů představovaly testy původně určené pro cizojazyčné nebo neslyšící pacienty. V dnešní době se však o mozku ví mnohem více informací než tehdy a mnoho dřívějších postupů v rámci neuropsychologie muselo být zcela revidováno. [7]

3.3 Chyby v měření při testování

Ať už se vědci v oblasti psychologických testů snaží sebevíc o vytvoření testů, které budou schopné změřit konkrétní psychické procesy přesně a objektivně, každý test má větší či menší odchylku výsledků od skutečnosti. Jsou testy, které jsou známy pro svoji vyšší přesnost a také testy, u kterých je správnost výsledků spekulativní.

Odchylku od skutečnosti způsobují chyby měření, které lze rozdělit do čtyř kategorií. [6]

3.3.1 Konstantní chyby

Vzhledem ke skutečnosti, že se psychické procesy nedají měřit přímým způsobem, můžeme pouze spekulovat o tom, jestli testy měří opravdu ty konkrétní věci, které by měřit měly. Jeden z problémů může nastat, když člověk vyplňující písemný test na paměť je dysgrafik, a mimo jiné se u testu měří i doba zpracování. V tomto momentě je výsledek ovlivněn nejen rychlostí vybavování jedince, ale i jeho zhoršenou schopností psát. [6]

Jako *validita testu* nazýváme soustavu důkazů o tom, jestli test měří co má a pokud ano, tak do jaké míry. Existují tři zdroje důkazů o validitě: [6]

- **Obsahové zdroje** se točí okolo obsahové reprezentace testované domény. Aby výsledné skóre testu správně reprezentovalo testovanou oblast, musí test obsahově spadat pod doménu testované oblasti. [5]
- **Empirické zdroje** dokazují určitou schopnost testu predikovat. V případě, že chci vytvořit empirické zdroje k přijímacímu testu na vysokou školu, vypočtu si, jak koreluje výsledné skóre testu u jednotlivých studentů s jejich studijním průměrem. Studijní průměr je v tomto případě tzv. *kritérium*. Čím větší je korelace s kritériem, tím více je test validní v tomto ohledu. [5]
- **Konstruktové zdroje** se určují v případě, že měříme něco, pro co se nedá jednoduše určit odpovídající kritérium. Tím něčím může být například inteligence. Inteligence je psychologický konstrukt, fyzicky neexistuje, nemůže tedy být kritériem. V takovém případě

je třeba konstrukt řádně definovat a určit vztahy tohoto testu s ostatními příbuznými testy. [5]

3.3.2 Osobní chyby

Osobní chyby může do testu vnést osoba, která test administruje, vyhodnocuje, případně interpretuje. Špatným administrováním testu může dojít u probanda k tomu, že test špatně pochopí a výsledky pak budou zkreslené. Zkreslit měření může také nevhodné prostředí — například přílišná zima či horko v místnosti, kde se testuje. Při interpretaci výsledků mohou dva různí lidé dojít k dvou různým závěrům. [6]

Měřítkem osobních chyb je tzv. *objektivita testu* a jeden ze způsobů, jak objektivitu zajistit, je standardizace testu, tedy určení srozumitelných pravidel administrace, vyhodnocování a interpretace výsledků. [6]

3.3.3 Proměnné chyby

Proměnná chyba nastává v případě, že výsledky při opakovaném měření stejné osoby jsou zcela odlišné. Dá se přirovnat k proměnné chybě ve fyzikálním měření, kdy opakovaným měřením stejného objektu dospějeme vždy k trochu jiné hodnotě. Měřítkem proměnných chyb je tzv. *reliabilita testu*. Existuje několik způsobů, jak odhadnout, jak moc je test reliabilní. [6]

- **Stabilita v čase** nebo-li testová-retestová reliabilita se pokouší odhadnout reliabilitu v případě, kdy test administrujeme ve dvou různých časech. Tento způsob odhadu spolehlivosti testu není vhodný pro druhy testů, které měří vlastnost, která se v čase přirozeně mění. Příkladem správného použití je u testu inteligence. Proband vyplní test jednou a poté po určité uplynuté době stejný test vyplní po druhé. Reliabilitu zjistíme tak, že vypočteme korelaci výsledků z prvního provedení testu s výsledky z druhého provedení stejného testu. [5]
- **Metoda paralelních forem** se použije k určení reliability v případě, že chceme vyloučit možnost, že si proband při provádění druhé verze testu bude pamatovat předešlé výsledky. Při této metodě existují dvě verze jednoho testu, každá verze má trochu jiné testovací položky. Opět se počítá vzájemná korelace výsledků obou testů. [5]
- **Metoda split-half** rozdělí test na dvě poloviny s tím, že u každé poloviny se určuje skóre zvlášť. Mezi výsledným skóre obou polovin se poté vypočte korelace pomocí speciálního Spearmanova-Brownova vzorce, který vypočte korelaci obou polovin tak, jakoby každá polovina představovala délku celého testu. [5]

3.3.4 Interpretační chyby

Tyto chyby vznikají v případech, kdy je výsledné skóre testu interpretováno v nevhodném kontextu. Protože nemáme u psychologických testů určenou jednotnou stupnici ani jednotku hodnot,

které měříme, nemůžeme srovnávat hrubé skóre dvou různých testů mezi sebou. Jedna z možností jak srovnat skóre dvou odlišných testů, je převést hrubé skóre na tzv. *interaktivní skóre*, které nabývá 0 – 100%, kde 0% představuje nejnížší dosažitelné skóre daného testu a 100% nejvyšší dosažitelné skóre. [6]

Na základě interaktivního skóre však nejsme zcela schopni srovnávat skóre různých osob. K tomu je třeba, aby se srovnávaly skóre osob vždy ze stejné referenční skupiny. Referenční skupinou rozumíme skupinu lidí se stejnými vlastnostmi definovanými v dané referenční skupině, například věk, pohlaví, zaměstnání atd. [6]

Vytváření těchto referenčních skupin je označováno jako tzv. *normalizace*. [6]

4 Testování kognitivních funkcí s využitím počítačů

Počítače se v dnešní době využívají téměř v každém oboru lidské činnosti. Psychologie není výjimkou. Psychologické testy z velké části představují činnosti, u kterých není problém jejich nahrazení formou počítačového programu. Všechny psychologické testy psané na papíře se dají zdigitalizovat, počítače nemají problém ani s jejich vyhodnocením, pokud se jedná o čistě matematické operace. Psychologické testy užívající fyzických předmětů lze předělat do grafické aplikace. Samozřejmě testy vyžadující psychologovu zkušenost a odborný úsudek je problematické nahradit jejich počítačovými variantami, jelikož umělá inteligence dnes není na takové úrovni, aby byla schopna nahradit osobu psychologa.

4.1 Typy využití

V praxi se počítače používají v rámci psychologického testování různými způsoby.

Mohou se používat jako platforma nesoucí samotný test. Zastupují tak funkci jak administrace testu tak i jeho vyhodnocení. Ukázalo se, že reliabilita většiny digitalizovaných psychologických testů se liší od jejich papírové podoby minimálně. Užití internetu navíc poskytuje psychologům možnost nasbírat obrovské množství dat. Něco, co by u testů v papírové formě zabralo mnohem více času a zdrojů. [5]

Další využití tkví ve sbírání dat o pacientech. Počítač může sloužit jako prostředník mezi pacientem a psychologem. Pacient své psychologické problémy shrne počítači a sesbíraná data pak vyhodnotí psycholog. Tímto způsobem dokáží počítače ušetřit psychologům čas. [5]

Lze také naplno využít počítačových technologií a zkonstruovat testy, které by bez počítačů nemohly existovat. Skupina vědců vyvinula test na měření kognitivních funkcí v každodenním životě. Tento test probíhá ve virtuální realitě ve virtuálním nákupním středisku. Testovaní lidé mají za úkol provádět různé úkoly týkající se nakupování. [9]

Virtuální realita také umožňuje navodit různé nereálné nebo nebezpečné situace. Používají se testy ve virtuální realitě k měření fyziologických reakcí (například tepu) u osob, které trpí fobiemi. Prostředí ve virtuální realitě pak může představovat palubu letadla, terárium s jedovatými pavouky nebo třeba pódium před zaplněným sálem. [5]

Jako další typ se mohou brát počítačové adaptivní testy. Tento druh testu je schopen dynamicky vyhodnocovat, jak obtížné mají jednotlivé položky testu být, na základě předešlých odpovědí pacienta. Výhoda těchto testů je menší počet položek nutných k vypočtení přesného skóre. Nevýhoda je mimo jiné nutnost, aby pacient znal základy práce s počítačem. [5]

Jako poslední lze uvést použití pro měření rychlosti reakcí v psychofyzice. Na obrazovce se na krátký moment zobrazí určitý signál a pacient na něj musí v co nejkratší době zareagovat. Existuje mnoho variací tohoto druhu testů, dá se měnit délka zobrazení signálu, počet signálů, nebo je třeba možné zobrazit v krátkém sledu dva signály s tím, že jeden signál slouží pouze ke zmatení pacienta. [5]

4.2 Výhody a nevýhody

Užití počítačů v psychologii s sebou nese jako v jiných oblastech řadu výhod a nevýhod.

V roce 1966 zjistil Dr. Joseph Weizenbaum, že jeho pacienti nemají problém svěřovat se se svými psychickými problémy počítačovému programu. Dokonce v některých případech bylo zjištěno, že se lidem mnohem lépe formulovali jejich problémy počítači, než samotnému terapeutovi. Důvodem je, že lidé mají v případě počítačového programu menší obavy při sdílení citlivých informací. [5]

V případě srovnávání reliability a validity testů v jejich klasické podobě a testům na počítači se ukázalo, že jsou tyto hodnoty téměř totožné. V některých případech je reliabilita stejných testů na počítači dokonce vyšší. [5]

Testování pomocí počítačů také zvyšuje objektivitu testu. Počítač se nemůže splést při vyhodnocování výsledků, instrukce podá vždy stejným způsobem a neunaví se. Šetří také čas psychologa, jelikož testy dokáže i vyhodnotit. [5]

Na druhou stranu testování pomocí počítačů vyžaduje od testovaných osob schopnost s počítačem pracovat — jejich úroveň schopností ovládat myš nebo například psát na klávesnici může ovlivnit jejich výkon. [5]

V případě větší znalosti fungování testovacího programu lze i podvádět a dosáhnout tak většího skóre. Tady už však záleží na konkrétních případech programů testujících kognitivní funkce.

Nemožnost v dnešní době zastoupit funkci psychologa počítačem se dá považovat jako další nevýhoda. [5]

4.3 Příklady aplikací měřících kognitivní funkce

Mezi české aplikace patří například NELA. NELA je webová aplikace, která umožňuje lékařům neurokognitivní laboratoře Neurologické kliniky Fakultní nemocnice Ostrava do systému zaregistrovat pacienty, kteří pak mohou provádět psychologické testy. Výsledky svých pacientů si lékaři mohou zobrazit v podobě grafů.

Kognitivní funkce, které NELA měří, jsou pozornost a paměť. Pozornost je měřena pomocí Bourdonova testu, v rámci kterého je pacient donucen vybírat z náhodné množiny tvarů tvary takové, které se vyskytují v předloze (viz obrázek 3). V tomto testu se měří obzvlášť koncentrovaná pozornost, přesnost percepce a psychomotorické tempo. [10] NELA byla vyvíjena studenty VŠB-TUO v rámci několika diplomových prací.

V zahraničí existuje aplikace Brain on Track. Tento test kognitivních funkcí se skládá z devíti subtestů, které měří pozornost, exekutivní funkce, jazykové funkce, prostorové vnímání a konstruktivní myšlení. Testy v sobě obsahují prvek náhody k zajištění větší reliability — lidé, kteří test vykonávají vícekrát, nemohou spoléhat na to, že si předešlé výsledky zapamatují. Rovněž bylo vytvořeno několik verzí obtížnosti k adaptivnímu testování.



Obrázek 3: NELA – Bourdonův test

Před vykonáním každého subtestu se pacientovi přehrají namluvené instrukce a poté je spuštěn časový limit dvou minut. Počet úloh v rámci jednoho subtestu je limitovaný pouze časovým limitem. Skóre každého subtestu odpovídá počtu správně vyřešených úloh subtestu. Test Brain on Track byl administrován pacientům s mírnou kognitivní poruchou a pacientům s demencí v raných stádiích. [11]

4.4 Výstupy do neurorehabilitačních aplikací

Cílem měření kognitivních funkcí může být zjistit, které kognitivní procesy je třeba trénovat, tedy určení obsahu následné neurorehabilitace. Různé aplikace mohou mít různé způsoby hodnocení těchto kognitivních procesů, například NELA měří pozornost a paměť.

Mezi výhradně neurorehabilitační aplikace patří například český Mentem, webová aplikace, která umožňuje trénovat různé oblasti kognitivních funkcí, například pozornost, paměť, a rychlost reakcí. Trénování probíhá formou her, které jsou šité na míru jednotlivým kognitivním procesům. [12]

Pokoušel jsem se vyhledat aplikaci, která by umožňovala analýzu kognitivních funkcí a následně jejich neurorehabilitaci, žádnou jsem však nenašel. U čistě neurorehabilitačních aplikací je bez znalosti jejich fungování těžké určit vhodné výstupy, které by mohly administrovat úlohy na míru.

Co se týče projektu Eddie, v rámci kterého bude nasazen diagnostický nástroj, který je předmětem této bakalářské práce, výstupy z diagnostického nástroje budou sloužit pro určení

neurorehabilitačních úloh. Probíhá zde však interní komunikace lidí, kteří v tomto projektu pracují, výstupy z diagnostického nástroje jsou tedy známy lidem, kteří budou neurorehabilitační úlohy implementovat.

5 Webové aplikace

Webovou aplikací rozumíme počítačový program v architektuře klient-server, využívající protokolu HTTP k přenosu dat mezi klientem a serverem v rámci internetu. Server odesílá klientu výstup, hypertextový dokument, který vznikne při kompilaci zdrojového kódu aplikace. Klientem bývá webový prohlížeč, který data přicházející ze serveru přečte a případný hypertextový dokument pak zobrazí v uživatelsky přijatelné podobě. [13, 14]

Webové aplikace nejsou ekvivalentní webovým stránkám. Webová stránka je jednoduchý statický dokument, který není třeba žádným způsobem kompilovat. Příkladem může být jednoduchá prezentace propojená hypertextovými odkazy. Webová aplikace v sobě obsahuje určitou funkcionalitu, kterou by samotný HTML kód obstatat nedokázal. HTML totiž není programovacím jazykem. Sociální sítě, e-shopy, webová rozhraní e-mailových schránek a například blogy jsou všechno webové aplikace. [15]

Při srovnání počítačových programů instalovaných a spouštěných na klientském počítači a webových aplikací, můžeme dojít k několika patrným rozdílům, které dělají tyto dvě různé technologie více či méně vhodné pro konkrétní užití. Mezi výhody webových aplikací například patří: [13]

- Data, se kterými uživatel pracuje, nejsou uložena na jeho počítači, nýbrž na externím serveru. Dá se k nim tedy dostat z jakéhokoliv zařízení.
- Vývojáři nemusí vyvíjet webovou aplikaci zvlášť pro více platform.
- Nové verze webové aplikace nemusí být distribuovány mezi uživatele, stačí novou verzi nahrát na webový server a uživatelům se při dalším požadavku na web zobrazí aktuální verze.

Naopak mezi jejich nevýhody patří:

- Výkon aplikace z uživatelského hlediska závisí zvláště na kvalitě připojení k internetu.
- I přes dříve zmíněnou kompatibilitu s různými platformami, vzhled uživatelského rozhraní webových aplikací a funkcionalita ze strany klienta závisí na konkrétním webovém prohlížeči a zařízení, na kterém je webová stránka zobrazována.
- Vzhledem k tomu, že při používání webových aplikací dochází k přenosu dat na server a zpět, klesá výkonnost webových aplikací rapidně s náročností jejich využití — webová aplikace k editaci videa bude rozhodně pomalejší, než software nainstalovaný přímo na klientském počítači.

5.1 HTTP protokol

Základní principy fungování webových aplikací i jednoduchých webových stránek spočívají v protokolu HTTP.

Tento komunikační protokol určuje pravidla pro komunikaci mezi webovým prohlížečem a webovým serverem. V rámci těchto pravidel je definována syntaxe zpráv, které umožňují serveru oznámit, jaký hypertextový dokument má klientovi (většinou webový prohlížeč) poslat včetně samotného způsobu odesílání. Když zprávy odesílá klient serveru, říká se, že posílá na server *požadavek*. Server pak klientovi *odpovídá*.

Požadavek odeslaný na server může vypadat takto:

```
GET / HTTP/1.1
Host: www.seznam.cz
Accept-Language: cs-CZ
Accept: text/html
```

Výpis 1: HTTP požadavek na hlavní stránku webu seznam.cz

Slovo *GET* značí operaci, kterou chce klient na serveru provést, v tomto případě se jedná o získání hypertextového dokumentu. Mezi další operace se řadí například *POST*, *PUT*, *HEAD* a *OPTIONS*. Za operací následuje adresa dokumentu, ke které se operace váže. V našem příkladě je to kořenový dokument nebo-li hlavní strana webu (/). Dále se jedná o určení verze protokolu HTTP (1.1).

V další části jsou nepovinné informace, tzv. *hlavička*, které mohou specifikovat adresu webu, na které je dokument uložen (www.seznam.cz), jaké jazyky prohlížeč upřednostňuje (cs-CZ), jaký druh dokumentu očekává (text/html) a další.

Pokud serveru, který je specifikován v požadavku, zpráva přijde, odpoví:

```
HTTP/1.1 200 OK
Date: Fri, 05 Apr 2019 17:29:56 GMT
Content-Type: text/html
```

Výpis 2: HTTP odpověď z webu seznam.cz

Mimo dříve zmíněnou verzi protokolu se zde objevuje také stavový kód, který určuje, jestli byl požadavek úspěšný, případně k jaké chybě došlo. *200 OK* značí, že požadavek byl úspěšný. Další stavové kódy jsou například *401 Unauthorized* (klient není autorizovaný k přístupu), *404 Not Found* (dokument nebyl nalezen), *500 Internal Server Error* (chyba serveru). Dále opět následují informace v hlavičce a za hlavičkou samotný obsah dokumentu, tedy v nejčastějším případě HTML kód. [16]

5.2 Technologie webových aplikací na straně klienta

Vzhledem k celkovému fungování webových aplikací jako architektura server-klient, lze technologie webových aplikací rozdělit do dvou kategorií. První z nich jsou technologie na straně klienta.

Skripty vytvářené v rámci těchto technologií se spouštějí pouze na klientském počítači. Přináší to tak výhodu v rychlém provádění operací, které ke svému chodu nepotřebují komunikovat přes internet. Zároveň se takto rozděluje zátěž webové aplikace a nemusí se o vše starat technologie na straně serveru. Nevýhodou (v některých případech i výhodou) může být možnost uživatele kdykoliv nahlédnout do zdrojového kódu. [18]

Mezi tyto technologie patří například HTML, CSS a JavaScript.

5.2.1 HTML

HTML je značkovací jazyk, který je základem každé webové stránky. Při požadavku webového prohlížeče na webovou stránku je součástí odpovědi serveru právě HTML dokument. HTML dokument pak v sobě dále odkazuje na další soubory (CSS, JavaScript), které webová stránka vyžaduje ke svému správnému zobrazení. [18, 19]

Význam HTML spočívá v hypertextových odkazech, které směřují na další webové stránky a umožňují tak uživateli procházení webem. Základem HTML syntaxe jsou tzv. *značky* (nebo častěji používaný anglický výraz *tags*), které ohraničují HTML elementy. Element je například odstavec textu, nadpis, obrázek nebo hypertextový odkaz. Pomocí těchto tagů určuje HTML strukturu webových stránek. [18, 19]

Existují párové a nepárové tagy. Párový tag se skládá ze dvou značek, které obklopují část webového obsahu. Nepárový tag je jediná značka, příkladem může být tag na vložení obrázku.

```

```

Výpis 3: Ukázka HTML elementu img

Každý tag může mít definovaný atribut. Tag pro vložení obrázku ve výpisu 3 obsahuje atribut, který určuje adresu umístění obrázku (src) a atribut title, který v sobě nese popis, co na daném obrázku je, užitý v případě špatné url či při prohlížení webové stránky zrakově postiženým člověkem.

HTML umožňuje ve svém kódu použití i jiných webových technologií, například určit vzhled stránky pomocí CSS přímo v atributu HTML elementu. Nicméně je doporučováno vyhnout se takovému řešení, nechat HTML na starost pouze strukturu webu a zbytek přenechat ostatním webovým technologiím. Ulehčuje se tímto údržba kódu a naplňuje princip oddělení zodpovědností. [18]

5.2.2 CSS

CSS nebo-li kaskádové styly umožňují definovat vzhled a rozložení HTML elementů. Vzhled definují pomocí vlastností (barvy, velikosti, způsob zobrazování atd.), které se vážou k danému HTML elementu pomocí *selektorů*.

```
img {  
  border: 2px solid blue;  
}
```

Výpis 4: CSS selektor a tělo s vlastnostmi

Selektor v našem příkladě je vázaný přímo na element `img` a určuje jedinou vlastnost, a to modrý, dva pixely široký rámeček okolo obrázku (HTML element `img`).

Selektory mají určitou specifičnost — vlastnosti v selektoru s vyšší specifičností přepíší vlastnosti v selektoru s nižší specifičností. Například selektor vázaný na atribut `id` HTML elementu bude mít vyšší specifičnost, než selektor, který se váže na třídu elementu. Tomuto přepisování vlastností se říká *kaskáda*. Kaskáda ovšem nevzniká pouze z důvodu specifičnosti selektorů, vliv má také pořadí selektorů v kódu a klíčové slovo *!important*.

CSS postrádá mnoho výhod klasických programovacích jazyků, například není v něm možné používat podmínky a cykly. Proměnné se v CSS vyskytují až v novějších specifikacích, které mnoho prohlížečů v době psaní této práce nepodporuje. [20] Je však možné použít tzv. *CSS preprocesory*, které poskytují možnost noření selektorů, tvorbu proměnných, cyklů a podmínek. Součástí preprocesorů je i jejich kompilátor, který kód zkompileje do klasického CSS. Příklady CSS preprocesorů jsou SASS a LESS. [18]

5.2.3 JavaScript

JavaScript je dynamický, interpretovaný programovací jazyk. Umožňuje vývojářům webových aplikací přidat na webovou stránku prvek větší interaktivity. Vznikl v rámci specifikace ECMAScript. [18]

Pomocí JavaScriptu lze manipulovat se strukturou HTML kódu a provádět asynchronní požadavky na různá webová API. Výhoda asynchronního dotazování spočívá v tom, že uživatel nemusí znovu načítat webovou stránku pro to, aby se mu zobrazila nová data. Právě díky asynchronnímu dotazování se dají vytvářet tzv. SPA aplikace, tedy Single Page Applications — aplikace v rámci jediné webové stránky. [18]

Pro JavaScript existuje několik frameworků a knihoven. jQuery je velmi často používaná knihovna zvláště z důvodu jednoduchosti použití. Mezi frameworky se řadí Angular, který funguje na vyšší úrovni abstrakce než základní JavaScript a vyžaduje proto také méně kódu. React je knihovna, která se vyznačuje svým virtuálním DOM a vkládáním HTML přímo do svého kódu. [18]

5.3 Technologie webových aplikací na straně serveru

Technologie na straně serveru představují programovací jazyky, které umožňují zpracování požadavku z uživatelského prohlížeče a vytvoření adekvátní odpovědi.

5.3.1 PHP

PHP je skriptovací jazyk na straně serveru. Umožňuje vytváření dynamických webových stránek, jinak řečeno webových aplikací. V momentě, kdy na server, kde funguje PHP, dojde z prohlížeče požadavek na zobrazení konkrétní webové stránky, PHP kód je interpretován a výsledkem je HTML kód. Zdrojový kód tedy není nutné, na rozdíl od jiných jazyků, kompilovat. [21]

PHP se často používá v kombinaci s webovým serverem Apache, databází MySQL a operačním systémem Linux — tzv. sada softwaru LAMP.

V dnešní době je to populární technologie na vývoj webových aplikací. Velké množství webů je postaveno pomocí Wordpressu, což je oblíbené CMS, které je vyvíjeno právě v PHP. [22]

5.3.2 Node.js

Node.js je běhové prostředí, ve kterém je možné spouštět JavaScriptový kód na straně serveru. Výhodou Node.js je, že neblokuje I/O operace, například při čtení z databáze zbytečně neplýtvá procesorovými cykly čekáním, místo toho pokračuje ve vykonávání následujícího kódu a na přerušenou operaci naváže až při odpovědi databáze. Tímto je Node.js schopen zvládat několik souběžných operací naráz i přes absenci vícevláknového vykonávání kódu. [23]

Node.js má také k dispozici velké množství rozšiřujících knihoven, které se dají spravovat pomocí Node.js package manageru. [23]

5.3.3 Java webové technologie

Webové aplikace vyvíjené v Javě obvykle nejsou spouštěny přímo na serveru. Místo toho je se serverem spojuje tzv. webový kontejner, který jim poskytuje běhové prostředí. Jedná se o analogii ke klasickým lokálním Java aplikacím, které jsou spouštěny v Java Virtual Machine. [24]

Obvyklá Java webová aplikace v sobě obsahuje servlety (třídy, které zpracovávají HTTP požadavky a odpovědi) nebo JavaServer Pages, které umožňují vkládat Java kód přímo do HTML kódu. [24]

Oblíbenými webovými frameworky v rámci Javy jsou Spring, JSF a například GWT. [24]

5.3.4 ASP.NET

ASP.NET je framework, který rozšiřuje .NET platformu od Microsoftu. Je postaven stejně jako samotná .NET platforma na Common Language Runtime, což přináší výhody jako je například Garbage Collection, typová kontrola a nebo dědičnost. ASP.NET vznikl z předešlého ASP. Jeden z rozdílů při srovnání s předešlou verzí je kompilace kódu namísto interpretace. [25]

Umožněna je jednoduchá tvorba webových aplikací implementujících návrhový vzor Model-View-Controller. Šablony webových stránek lze vytvářet pomocí Razor šablonového enginu, který umožňuje použití programovacího jazyka C# přímo v HTML kódu. [26]

Aktuálně se ven dostává nová verze, tzv. ASP.NET Core, open source varianta klasického ASP.NET s podporou více platforem. V ASP.NET Core je tedy možné nasadit aplikaci i na jiných operačních systémech. [26]

6 Ochrana a archivace dat

V dnešní době, kdy má téměř každý člověk umožněn přístup k internetu a možnost vytvořit si vlastní účet na několika různých sociálních sítích, je čím dál menším problémem dohledat si informace téměř o komkoliv. Mnoho lidí však dnes využívá i internetových bankovníctví či zadává citlivé informace, které jsou pak ukládány do databází, mnohdy v podobách, kdy by únik těchto dat mohl představovat problém.

Mezi tyto citlivé informace se řadí i patientská data, která mohou být shromažďována různými zdravotními institucemi nebo jinými subjekty.

6.1 GDPR

V roce 2016 bylo Evropským parlamentem a Radou EU vydáno obecné nařízení o ochraně osobních údajů (v originále General Data Protection Regulation, tzv. *GDPR*). Oficiální název je *Nařízení Evropského parlamentu a Rady 2016/679 ze dne 27. dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES*, účinnost tohoto nařízení nabyla 25. května 2018. [27]

GDPR přineslo řadu změn týkajících se, zjednodušeně řečeno, zpracovávání a uchovávání osobních údajů v Evropské unii a výstupem těchto dat mimo EU. Vzhledem k tomu, že osobní data jsou spravována velkým množstvím lidí, zasáhly tyto změny v podstatě celý svět. Řada amerických zpravodajských portálů byla vynucena zamezit přístup uživatelům vyskytujícím se v Evropské unii, jelikož webové stránky těchto portálů nebyly v tu dobu připraveny splňovat toto nařízení. [28]

GDPR v sobě specifikuje *práva subjektu údajů a povinnosti správce údajů*. Mezi práva subjektu údajů se mimo jiné řadí: [27]

- Právo být transparentním, srozumitelným a snadno přístupným způsobem informován o osobních údajích, které byly získány s i bez svého souhlasu.
- Právo na přístup ke svým osobním údajům. To znamená, že správce musí na žádost subjektu údajů vydat kopii zpracovávaných údajů.
- Právo na opravu osobních údajů či doplnění neúplných údajů.
- Právo na výmaz, které pro správce údajů představuje nutnost osobní údaje subjektu bez odkladu vymazat v případě, že je naplněn jeden ze specifikovaných důvodů (např. osobní údaje už nejsou potřebné pro účely, které byly shromážděny nebo subjekt údajů odvolá souhlas k poskytnutí osobních údajů).

Správce má povinnost zpracovávat osobní údaje v souladu se zásadami GDPR. Mezi tyto zásady patří: [27]

- Zpracovávat osobní údaje subjektu jen v případě, že subjekt udělil souhlas.

- Zpracování osobních údajů musí mít účelové omezení. Je tedy nutné, aby správce předem specifikoval, k čemu tato data potřebuje a následně je používal jenom tímto způsobem.
- Osobní údaje je možné spravovat jen do doby naplnění účelu, pro který byla data shromažďována.

6.1.1 Anonymizace dat

Anonymizace je proces, který se úzce vztahuje s GDPR a ochranou osobních údajů. GDPR specifikuje, že data, která již nenaplnují svůj účel, je třeba anonymizovat.

Anonymizace probíhá tak, že se z dat odstraní všechny informace, které by mohly data spojovat s konkrétní osobou. Mohou to být jména osob, adresy jejich bydliště, telefonní čísla, e-mailové adresy a podobné údaje. Kromě přímého odstranění lze také data v digitální podobě zašifrovat a klíč k vzniklé šifře nikde neukládat.

Nicméně objevuje se zde ještě jeden termín podobný anonymizaci dat, tzv. *pseudonymizace* osobních údajů. Podle GDPR je nutné, aby všechna ukládaná data byla pseudonymizována. Znamená to, že data musí být ukládána v takové podobě, že kdyby se k nim dostala neoprávněná osoba, nedokáže z nich nic vyčíst, hlavně je tedy nedokáže spojit s konkrétní osobou.

Pseudonymizace představuje podobný proces jako anonymizace. Jediný rozdíl je v tom, že pseudonymizace je vratným procesem. Může se tedy jednat o šifrování dat s tím, že klíč je někde uložen. [29]

6.2 Možnosti ochrany dat ve webových aplikacích

Pod ochranou dat v rámci webových aplikací se skrývá mnoho různých metod, a každá se svým způsobem stará o zajištění toho, aby se data nedostala do rukou neoprávněné osoby a aby zůstala zachována. Tyto metody se dají rozdělit na procesy, které musí provádět samotný uživatel a procesy, které mají na starost lidé, kteří webovou aplikaci implementují a spravují.

Pacientská data, pokud jsou zpracovávána v rámci webových aplikací, se neliší od běžných dat. Snad je jen třeba říct, že se jedná o velmi citlivá data a proto je třeba zajistit jejich maximální ochranu.

Mezi jednotlivé způsoby ochrany dat patří těchto několik metod.

6.2.1 Šifrování dat

Při šifrování se data zakódují speciálním šifrovacím algoritmem do tvaru, který je nečitelný. K získání nějaké smysluplné informace je třeba data stejným algoritmem dešifrovat.

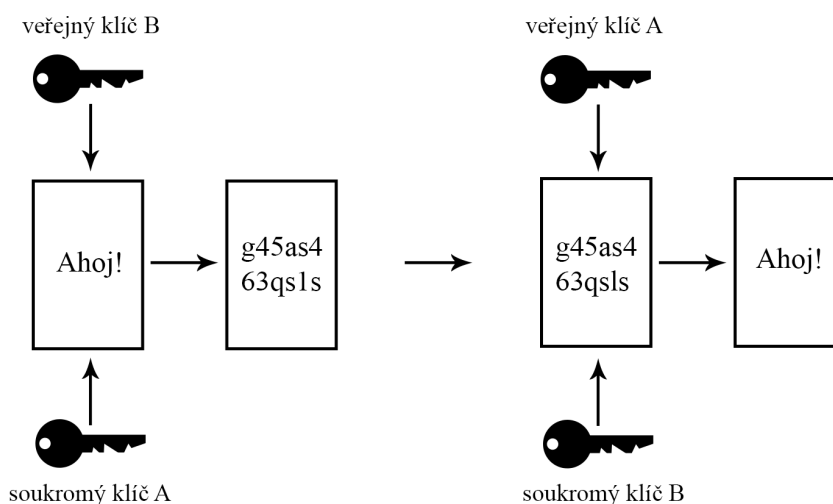
Tento způsob ochrany dat se používá v případě, kdy chceme, aby se k datům dostala pouze oprávněná osoba. Šifrování se však také používá zároveň k tomu, abychom se ujistili, že příchozí data opravdu pocházejí z místa, ze kterého by měla, jinak řečeno můžeme takto ověřit původ informace a potvrdit, že data nebyla na cestě k nám upravena. [30]

V dnešní době se šifrování velmi často používá k zabezpečení dat odesílaných mezi webovým serverem a prohlížečem, například bankovních údajů a dalších citlivých informací.

Tato zabezpečená komunikace probíhá skrz protokol HTTPS. Když uživatel odešle požadavek přes klasické HTTP, jsou údaje odesílány ve své nezměněné formě. Ty pak může člověk, který poslouchá spojení mezi uživatelem a serverem, přečíst. Při použití HTTPS jsou data odesílána v šifrované podobě, kterou dokáže rozluštit pouze cílový server. [31]

Existují dva základní způsoby, jak vyřešit šifrování dat:

1. **Symetrické šifrování** funguje tak, že strana, která data šifruje, odešle druhé straně soukromý klíč, který byl k zašifrování použit. Druhá strana následně může s pomocí tohoto klíče data dešifrovat. Symetrické šifrování je rychlé a často se proto používá na větším počtu dat, například k zašifrování databáze. [32]
2. **Asymetrické šifrování** je šifrování pomocí dvou veřejných a dvou soukromých klíčů. Celá funkcionality vychází z předpokladu, že data zašifrována pomocí veřejného klíče X lze dešifrovat pomocí soukromého klíče X a naopak.



Obrázek 4: Asymetrické šifrování

Když by chtěl uživatel A odeslat uživateli B nějaká šifrovaná data, zašifruje je pomocí veřejného klíče B, který je veřejně dostupný a pomocí soukromého klíče A, ke kterému má přístup pouze on. Jakmile uživateli B zašifrovaná data přijdou, dešifruje je pomocí svého soukromého klíče B a veřejného klíče A.

Data by šla v našem případě samozřejmě zašifrovat jen pomocí veřejného klíče, soukromý klíč však hraje důležitou roli potvrzení, že data přišla opravdu od uživatele A, nikdo jiný totiž přístup k soukromému klíči A nemá. [33]

6.2.2 Hashování

Na rozdíl od šifry je hashování jednosměrná funkce — k informaci, která byla zashashována, se člověk ze samotného hashe nedostane. Proto hashování shledává užití mimo jiné i v zabezpečení dat. [34]

Hash je vytvářen pomocí hashovacích algoritmů. Mezi tyto algoritmy patří: [35]

- **MD5**, hashovací algoritmus, který býval oblíbený, v dnešní době už je však nepoužitelný, jelikož byl prolomen.
- **SHA-1** býval používán na hashování dat až zhruba do roku 2010, kdy byly objeveny možné kolize hashů (různé vstupy produkují totožný hash). SHA-1 patří do rodiny SHA — hashovacích algoritmů vyvíjených americkou NSA.
- **SHA-2** je dnes bezpečnější variantou SHA-1, ovšem je očekáváno, že se brzy narazí na stejný problém jako u jeho předka.
- **SHA-3** je aktuálně nejbezpečnějším hashovacím algoritmem, je také značně rychlejší než předchozí varianty.

I přes to, že je hash jednosměrnou funkcí, je možné se k zahashované informaci dostat útokem hrubou silou (v angličtině tzv. *brute-force attack*). Princip útoku hrubou silou spočívá ve zkoušení všech možných kombinací. Při užití této techniky k odhalení zahashované informace útočník každou vzniklou kombinaci zahashuje a vzniklý hash srovná s hashem, který chce rozluštit.

Z důvodu větší bezpečnosti se proto doporučuje při hashování používat sůl, tedy náhodnou sekvenci znaků, která se připojí za hashovaný vstup. V případě, že útočník používá tzv. *slovníkový útok*, kdy místo náhodných dat generuje slova, která dávají smysl, dokáže takto osolený hash zmařit pokus o prolomení.

6.2.3 Zajištění zranitelných míst v aplikaci

Webové aplikace nejsou dokonalé, existuje mnoho skulinek, které představují nebezpečí v rámci ochrany dat. Nejčastěji se jedná o nevalidované vstupy skrze uživatelská rozhraní, případně využití slabiny ve webových technologiích na straně klienta.

- **Cross-site Request Forgery** představuje útok, který využívá skutečnosti, že si webová aplikace, kterou si útočník vybral, pamatuje přihlášení uživatele na uživatelské prohlížeči. Všechny požadavky na vybranou webovou aplikaci, které uživatel provede, tedy projdou, i ty, které se vyvolají kliknutím na odkaz, nebo načtením stránky s iframem, kde URL představuje požadavek. Pokud je cílová webová aplikace banka a požadavek představuje bankovní transakci na útočníkův účet, je to problém.

Řešením, jak se vyhnout CSRF, je ke každé HTTP odpovědi přidávat *CSRF token*, tedy vygenerovaný textový řetězec, který se uloží na serveru a poté je porovnán s tím v požadavku. V případě, že se nerovnájí, požadavek neprojde. [36]

- **SQL Injection** je útok, který útočníkovi umožňuje ovládat databázi webové aplikace — vypsat data, přihlásit se jako nějaký uživatel, změnit obsah databáze atd. SQLi využívá míst, kde aplikace přijímá uživatelská data, která následně používá uvnitř SQL příkazu.

V případě, že ve webové aplikaci máme přihlašovací formulář, kde je nutné vložit jméno a heslo, může útočník vhodně zvoleným textovým řetězcem vloženým do pole pro heslo, ošálit aplikaci tak, že ho přihlásí nezávisle na jeho znalosti hesla k účtu.

Pokud webová aplikace po požadavku přihlášení vyvolá SQL dotaz ve výpisu 5, a útočník do pole pro heslo vloží řetězec *0 OR 1=1*, upraví tím výsledný SQL dotaz tak, že ho aplikace přihlásí, i když heslo nezná.

```
string sqlCommand = "SELECT user_id FROM Users WHERE user_name=" + name +  
    " AND user_password=" + password;
```

Výpis 5: C# kód pro formátování SQL dotazu

Způsob, jakým se proti SQLi dá bránit, představují tzv. *parametrizované dotazy*. Parametrizované dotazy mají v sobě předem určená místa pro parametry, takže databáze je bude chápat jako data a ne jako část SQL příkazu. Další možností je tzv. escapování speciálních znaků v uživatelském vstupu, což představuje nahrazení speciálních znaků, které by mohly mít význam v SQL příkazu, znaky neškodnými. [37]

- **Cross-site Scripting (XSS)** je jeden z dalších možných útoků. XSS využívá webových stránek, kde mohou uživatelé vložit nějaký obsah, který se poté zobrazí v rámci dané stránky — mohou to být komentáře, profilové stránky, příspěvek na diskuzním fóru a další.

Útočník však jako obsah zvolí JavaScript, který se poté spustí v prohlížeči každému uživateli, jenž stránku se skriptem navštíví. Tímto způsobem může útočník získat kontrolu nad prohlížečem uživatele, dostat se k historii prohlížení nebo obsahu schránky operačního systému.

Pro zamezení tohoto druhu útoku je třeba podobně jako u SQLi escapovat uživatelův vstup do aplikace. [38]

6.2.4 Zálohování dat

Ani data, která jsou chráněná proti neoprávněnému přístupu, nejsou úplně v bezpečí. Kdykoliv se může přihodit nějaká situace, která způsobí, že data ze serveru ztratíme. Může to být selhání

hardwaru, chyba v aplikaci, korupce těchto dat a další podobné události. Jediný způsob, jak se proti nechtěné ztrátě dat bránit, je data pravidelně zálohovat.

Zálohování je proces, při kterém se vytváří kopie existujících dat, které jsou následně ukládány na externí server nebo lokální datový nosič. Proto se dají zálohy rozdělit na dvě kategorie — místní zálohy a tzv. off-site zálohy. [39, 40]

- **Místní záloha** se provádí na datový nosič v rámci jednoho datového centra. Datovým nosičem, na který se záloha ukládá, může být externí HDD, NAS nebo například magnetická páska. Přenos probíhá v rámci jednoho datového centra, je to proto bezpečná a velmi rychlá varianta zálohy. [40]

V případě zálohování databáze je první nutné tuto databázi přepnout do offline módu, takovému druhu místní zálohy se říká *studená záloha*. [41]

Do určité míry se jako záloha dá použít i technologie RAID. RAID funguje na principu několika redundantních disků. Existuje několik RAID úrovní. RAID 0 ukládá data po blocích na více disků (tzv. *proužkování*), umožňuje tím rychlejší přístup k datům, jelikož čtení může probíhat paralelně na více discích zároveň. RAID 1 automaticky duplikuje data mezi několika disků (tzv. *zrcadlení*), zajišťuje tím redundanci dat a tím snižuje pravděpodobnost jejich ztráty, kapacita paměti se ale s počtem disků nijak nenavýšuje.

RAID však není doporučován jako vhodná volba zálohování dat. V případě, že dojde ke korupci dat na jednom disku, automaticky se tato korupce zkopíruje na zbylé disky v RAID poli. Vzhledem k tomu, že se data ukládají do stejného stroje, v momentě fyzického poškození serveru je RAID k ničemu. Nejvhodnější je kombinovat RAID spolu s další metodou zálohování. [42]

- **Off-site záloha** probíhá na vzdálenou lokaci, kterou může být sekundární datové centrum společnosti, které data patří, nebo server vlastněný třetí stranou.

Případu provádění záloh na serveru třetí strany se říká skladování ve veřejném cloudu. Přináší to výhodu, že se společnost, které data patří, nemusí starat o správu těchto externích serverů. Nevýhodou je nutnost svěření dat do rukou třetí strany.

Skladování v soukromém cloudu představuje variantu zálohování dat na servery patřící společnosti, které data vlastní. [40]

Zároveň existuje několik metod, jak může záloha probíhat: [40]

- **Úplná záloha** je typická záloha, kdy se zálohuje vždy všechna data. Nevýhodou je však časová náročnost a požadavek většího počtu disků.
- **Inkrementální záloha** je záloha pouze těch dat, které se od předešlé úplné nebo inkrementální zálohy změnily. Tyto zálohy jsou rychlejší, ovšem obnovení dat je naopak zdoluhavým procesem.

- **Rozdílová záloha** ukládá data, která se změnila od poslední úplné zálohy. Přináší tak výhody obou předešlých způsobů. Záloha i obnovení je víceméně rychlý proces.

Vždy je třeba specifikovat, jak často mají zálohy dat probíhat. Nejideálnější je provádět úplné zálohy aspoň jedenkrát týdně, spíše o víkendech nebo v době, kdy data využívá nejmenší počet uživatelů. Zároveň je vhodné, aby několikrát týdně proběhla inkrementální nebo rozdílová záloha. [40]

6.2.5 Vyškolení uživatelů webové aplikace

Ať už se budeme snažit o zabezpečení dat ze strany databáze a aplikace sebevíc, stačí, aby uživatel zvolil příliš jednoduché heslo ke svému účtu a okamžitě tak vystaví svůj účet nebezpečí prolomení, nezáleží na tom, jak moc zabezpečenou aplikaci máme. Ještě horší situace by mohla nastat, kdyby zmíněným uživatelem byl člověk s administrátorskými právy. Pokud by měl správce v naší potenciální webové aplikaci přístup k uživatelským datům a někdo odhalil pomocí útoku hrubou silou jeho heslo, dojde k úniku těchto uživatelských dat.

Proto je třeba informovat každého člověka, který bude webovou aplikaci používat, o náležitostech bezpečného hesla a nejlépe rovněž při registraci omezit vkládané heslo minimální délkou a minimálním počtem speciálních znaků.

Snížit pokusy o prolomení uživatelských hesel se dá kromě omezení počtu špatných přihlášení také tak, že nebudeme veřejně informovat o tom, kdo je v naší aplikaci registrovaný. V případě, že člověk zadá do přihlašovacího formuláře jméno existujícího uživatele a špatné heslo, neohlásíme mu, že zadal špatné heslo, nýbrž že zadal špatné jméno *nebo* špatné heslo. Toto zamlčování uživatelů systému je samozřejmě obzvlášť klíčové v aplikacích, kde je tato informace citlivou, například ve webových aplikacích pracujících s patientskými daty.

7 Realizace webového portálu obsahujícího nástroj pro měření kognitivních funkcí

V rámci této bakalářské práce jsem měl za úkol implementovat webovou aplikaci, která bude umožňovat testování kognitivních funkcí svých uživatelů. Samotný testovací nástroj bude využit jako součást projektu Eddie, jehož cílem je neurorehabilitace pacientů se získaným poškozením mozku. Proto bylo třeba počítat i s tímto požadavkem, který však nebyl součástí zadání bakalářské práce.

Bylo nutné zkoordinovat vývoj aplikace s lidmi, kteří pracují na ostatních částech projektu Eddie, zvláště umožnit pozdější napojení diagnostického nástroje do tohoto projektu. Základní požadavky, jak by měla aplikace fungovat a vypadat, jsem shromáždil jak od vedoucího a konzultanta této práce, tak od psychologa PhDr. Petra Niliuse, Ph.D. U požadavků jsem zhodnotil jejich realizovatelnost, případně jsem domyslel, jak by konkrétní subtesty mohly fungovat.

7.1 Specifikace požadavků

Aplikace bude umožňovat zhodnocení kognitivních schopností uživatele formou dotazníku kognitivních chyb (CFQ) a diagnostického nástroje kognitivních funkcí — testu, skládajícího se z několika subtestů, zaměřených na konkrétní kognitivní funkce. Subtesty (v této práci používám také výraz *úloha*) budou řešeny formou jednoduchých her. Výsledek této analýzy kognitivních schopností bude možno použít jako vstupní data do navazujících částí projektu Eddie, specificky takto bude Eddie umět administrovat adekvátní neurorehabilitační úlohy.

Samotný proces testování bude probíhat takto:

1. Proveďte se jednoduchá anamnéza pacienta, tedy sesbírání osobních údajů. Konkrétně to bude věk, pohlaví a dosažené vzdělání. Tato data později umožní vyhodnocení výsledků pacienta v rámci odpovídající skupiny uživatelů.
2. Pacient vyplní dotazník CFQ.
3. Pacient provede test, realizovaný formou jednoduchých her. Test se bude skládat z několika subtestů, zaměřených na jednotlivé kognitivní schopnosti.
4. Pacient provede druhou verzi totožného testu k vyhodnocení reliability.
5. Pacientovi se zobrazí výsledky, data bude možné použít pro další práci v projektu Eddie.

Webová aplikace bude umožňovat registraci uživatelů, bude splňovat požadavky GDPR, uživatelská data budou dostatečně zabezpečena. Bude možné provádět zálohy a předešlý stav ze záloh obnovovat.

Mezi uživateli bude existovat správce s přístupem k anonymním uživatelským výsledkům. Tento správce bude mít možnost zobrazit si jednoduché statistické zpracování měřených dat a zhodnocení reliability testu spolu s vyhodnocením souvztažnosti dotazníku CFQ s testem.

Následuje stručná specifikace jednotlivých subtestů a také určení, na které kognitivní funkce se subtesty budou zaměřovat:

- První subtest bude zaměřený na vnímání a bude probíhat formou kategorizace zkráceně zobrazených objektů.
- V rámci subtestu zaměřeného na pamětní proces vstřípení označí uživatel objekty z předem daného subtestu.
- Dále bude následovat subtest zaměřený na rychlost reakcí. Uživatel bude mít za úkol co nejrychleji kliknout na náhodně zobrazované objekty.
- Další dva subtesty se budou týkat myšlení. Ten první bude zaměřený na schopnost plánování, obsahem budou Hanojské věže.
- Druhý subtest zaměřený na myšlení, konkrétně proces tvorby strategie, budou piškvorky. Tento subtest bude sloužit rovněž jako rozptýlení pro finální subtest.
- Cílem posledního subtestu bude znova označit objekty z prvního subtestu. Měřenou schopností je zde pamětní proces vybavování.

7.2 Návrh systému – webový portál

Po společné diskuzi s lidmi, pracujícími na ostatních částech projektu Eddie, jsme se dohodli, že k implementaci systému zvolíme ASP.NET. Webová aplikace, kterou v rámci této práce implementuji, tedy vznikla ve frameworku ASP.NET MVC. Jako databázi jsem z důvodu větší kompatibility zvolil Microsoft SQL Server.

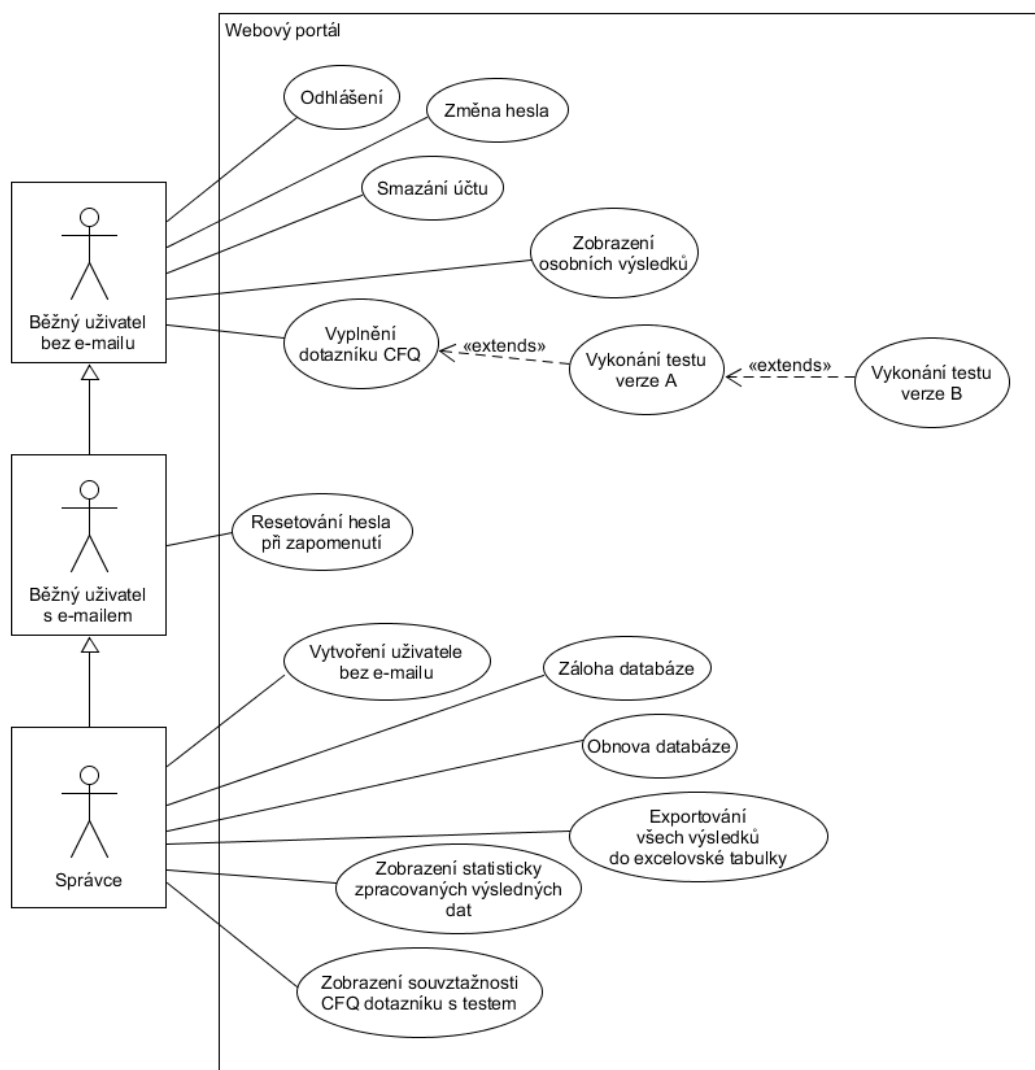
Rozhodl jsem se webový portál realizovat v podobě veřejně přístupné webové aplikace, do které se bude moci registrovat jakýkoliv uživatel.

Uživatelé systému jsem rozdělil na tři typy — běžný uživatel bez e-mailu, běžný uživatel s e-mailem a správce. Běžný uživatel se může registrovat samostatně, k identifikaci jsem se rozhodl využít e-mailovou adresu. Pokud tento uživatel zapomene přihlašovací údaje, může si nechat poslat na tento e-mail odkaz, který mu umožní resetovat si heslo.

Následně však přišel požadavek, aby bylo možné do systému dostat uživatele bez nutnosti použití e-mailové adresy v případě testování lidí, kteří e-mailovou schránku nemají. Do portálu jsem tedy pro správce přidal možnost přidávat uživatelské účty, které nejsou propojeny s e-mailovou adresou. Tito uživatelé ovšem nebudou mít k dispozici možnost heslo si resetovat.

Správce systému je uživatel, kterému byla přidělena správcovská práva. Při prvním spuštění webového portálu se přihlášenému uživateli na hlavní stránce zobrazí tlačítko umožňující do systému vložit správcovský účet, který má definované výchozí přihlašovací údaje.

Všechny případy užití jsem definoval v use case diagramu na obrázku 5.



Obrázek 5: Use case diagram webového portálu

7.2.1 Registrace uživatele a zabezpečení uživatelských dat

Jedním z dřívějších požadavků na systém bylo zajistit, aby si každý uživatel mohl zaregistrovat pouze jeden účet, na kterém bude test vykonávat. Cílem bylo vyhnout se situaci, kdy v databázi budou existovat dva výsledky jedné verze testu provedené jedinou osobou.

Splnění tohoto požadavku se však ukázalo jako nemožné v podobě, v jaké měl portál fungovat, tedy veřejně přístupná aplikace. Jediné informace, které se dají použít k identifikaci uživatele, jsou jeho IP adresa, MAC adresa, případně osobní informace, jako je například rodné číslo, číslo občanského průkazu a podobně.

IP a MAC adresy ovšem identifikují zařízení, ze kterého se uživatel registroval. Tuto kontrolu by se dalo obejít registrací z jiného zařízení, případně z jiné sítě či použitím VPN.

IP adresa by v případě, že by se jednalo o router poskytovatele internetu, identifikovala všechny klienty, vlastníci soukromou IP, jako jednoho uživatele. Výsledkem by bylo, že by se z této sítě mohl registrovat pouze jeden člověk. MAC adresa, tedy identifikátor síťového zařízení, je změnitelná.

Pokud bychom jednotlivé uživatele rozlišovali na základě rodných čísel, či jiných osobních údajů, nijak nezamezíme tomu, aby si je uživatel nevymyslel. Zpoplatnění registrace a následná identifikace podle čísla bankovního účtu či kreditní karty by odradila jen určitou část lidí od vytváření duplicitních účtů použitím druhé kreditní karty.

Jediným řešením tohoto problému by tedy bylo aplikaci používat interně a uživatele registrovat jako správce. Proto se nakonec od tohoto požadavku upustilo, rovněž také z důvodu, že se tyto problémy budou řešit až v rámci projektu Eddie, kde bude z výsledku této práce použit pouze diagnostický nástroj.

Registrace

E-mail

Heslo

Potvrzení hesla

Věk

Ukončené vzdělání

Vyberte ▼

Pohlaví

Vyberte ▼

Podmínky zpracování dat

Registrací souhlasíte s využitím svých dat k měření výsledků testů.

Konkrétně ukládáme Vaši e-mailovou adresu, heslo, věk, pohlaví, vzdělání, výsledek dotazníku, který vyplníte a výsledky testů, které provedete. E-mailová adresa je ukládána v nečitelné podobě a je využívána v případě zapomenutého hesla. Heslo je taktéž ukládáno v nečitelné podobě.

Ke svým výsledkům testů máte přístup pouze Vy a správce tohoto systému.

Kdykoliv se můžete rozhodnout svůj účet smazat a s ním i všechna Vaše data.

☐ Souhlasím

Registrovat

Obrázek 6: Registrační formulář

Při návrhu zabezpečení uživatelských dat jsem první zvažoval šifrování databáze. Později jsem si však uvědomil, že není třeba šifrovat všechna data, ale pouze zajistit, že data nepůjde

spojit s konkrétním člověkem. Webový portál využívá pouze jednu informaci, která by někoho mohla potenciálně identifikovat — e-mailovou adresu. E-mailová adresa je navíc využita pouze k resetování hesla v případě, že uživatel heslo zapomene. Vždy, kdy je e-mailová adresa v aplikační logice využita, se srovnává uživatelův vstup s databází.

Proto jsem se rozhodl, že e-mailovou adresu budu hned po registraci hashovat. Tím zajistím, že v případě, kdyby se obsah databáze z nějakých důvodů dostal ven, nebude možné e-mailové adresy přechýst a tím pádem ani spojit výsledná data s konkrétním člověkem. Jako hashovací algoritmus jsem zvolil SHA-2, který je v dnešní době postačitelý. Dalším údajem, který se hashuje je heslo a zde jsem k ještě většímu zabezpečení použil sůl.

Jako systém, který mi umožní implementovat registraci, přihlašování a identifikaci uživatelských účtů jsem zvažoval použít ASP.NET Identity. Ukázalo se však jako komplikované zajistit v tomto systému hashování e-mailových adres. Zvolil jsem tudíž Entity Framework a funkcionalitu registrace, odesílání potvrzovacího e-mailu, přihlašování a identifikaci uživatele jsem implementoval v rámci tohoto frameworku sám.

Samotný proces registrace jsem se snažil navrhnout tak, aby byl pro uživatele co nejjednodušší. Při registraci musí člověk vyplnit svoji e-mailovou adresu, heslo, věk, pohlaví a dokončené vzdělání (viz obrázek 6). Zároveň musí souhlasit s poskytnutím svých osobních dat k testovacím účelům. Tento krok je vyžadovaný při zpracovávání osobních dat v EU.

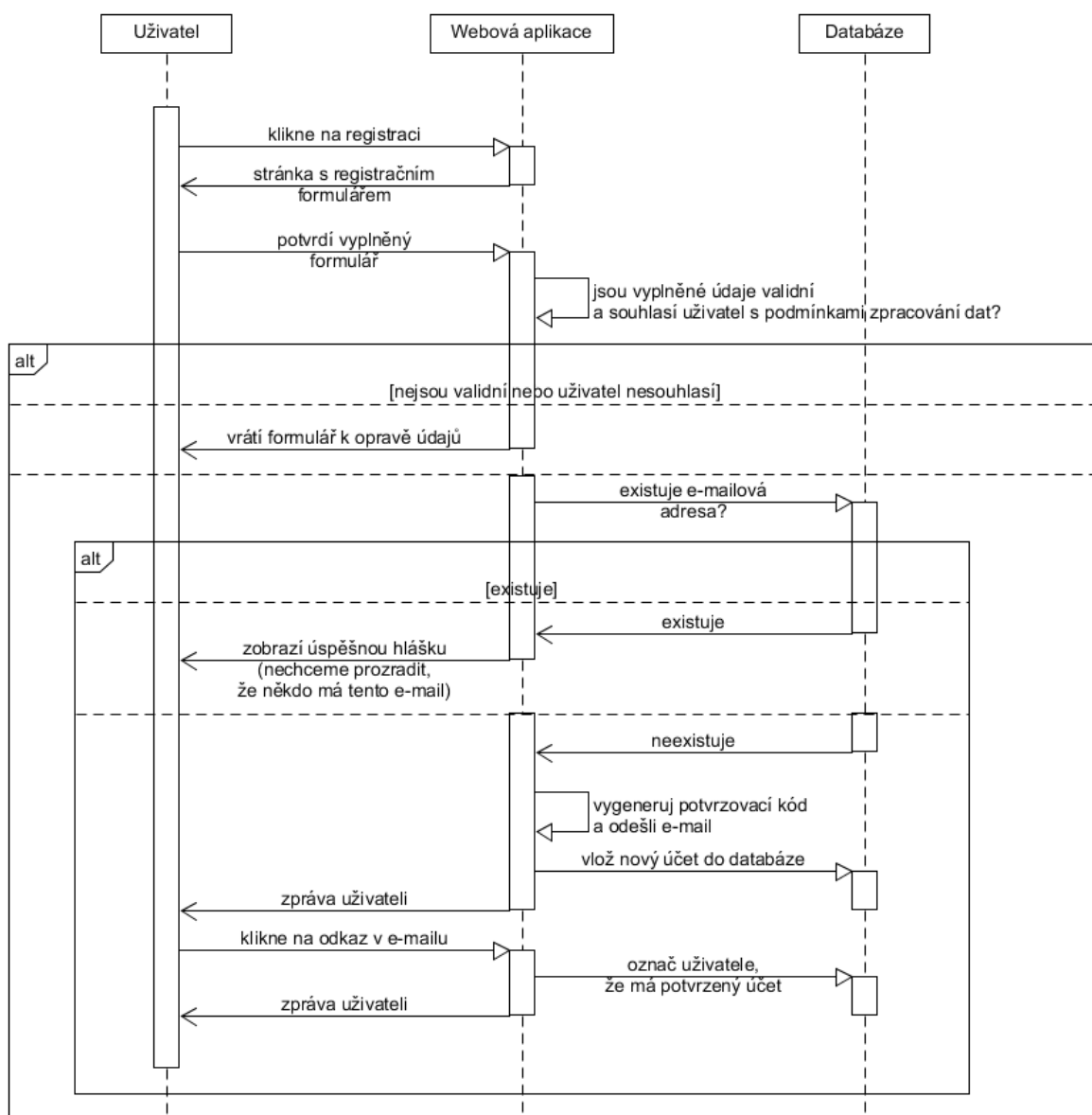
V případě, že e-mailová adresa již v systému existuje, uživatel se to nedozví. Tímto se zajistí anonymita registrovaných uživatelů. Jestli je konkrétní e-mailová adresa registrována v tomto systému je považováno za citlivou informaci, zvláště když se jedná o patientská data.

Po registraci přijde uživateli na zvolenou e-mailovou adresu e-mail, ve kterém se dozví své přihlašovací jméno a potvrzovací odkaz. Po kliknutí na odkaz se uživatelův účet aktivuje a uživatel se může přihlásit. V případě, že by svůj účet do jednoho dne nepotvrdil (třeba z důvodu, že zadal špatnou e-mailovou adresu a nemá, jak by se k potvrzovacímu odkazu dostal), uvolní se rezervace této e-mailové adresy v systému. Důsledkem je, že pokud by se později registroval někdo se stejnou e-mailovou adresou, předešlý účet se smaže.

K přihlašování a následné identifikaci uživatele v aplikaci využívám souborů cookies. V momentě, kdy se uživatel přihlásí, vygeneruje se unikátní textový řetězec, identifikační kód, který se uloží do databáze a do cookie souboru. Následně kdykoliv potřebuji zjistit, zdali je uživatel přihlášený, porovnáím identifikační kód v cookie souboru s kódem, který je v databázi. Tento identifikační kód zároveň používám při již zmíněném potvrzování/aktivaci účtu.

Jako jednou z podmínek GDPR je tzv. *právo byt zapomenut*. Uživatel má mít možnost odstranit svá data ze systému. V rámci tohoto webového portálu si uživatel může sám smazat svůj účet. Má však přitom na výběr, jestli chce zanechat svá anonymizovaná data — výsledky testu. V případě, že tak učiní, odstraní se z jeho účtu pouze hashované heslo a e-mailová adresa, zbytek zůstane zachován. Tato volba není ve výchozím stavu zaškrtnuta.

Na obrázku 7 se nachází sekvenční diagram podrobně znázorňující proces registrace.



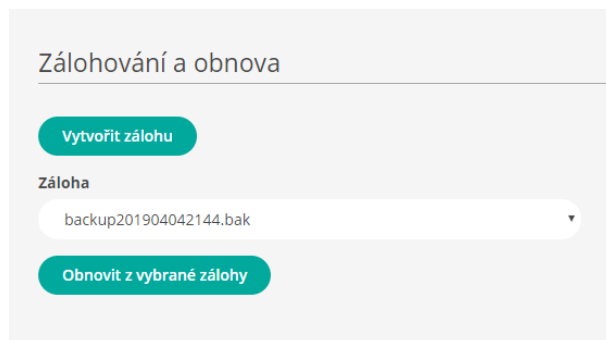
Obrázek 7: Postup registrace uživatele do systému

7.2.2 Zálohování dat

Přímo na webovém portálu je možné provádět zálohy a obnovení databáze, ovšem tato funkcionality je dostupná pouze v případě, že je webová aplikace spouštěna na stejném serveru, na kterém běží databáze. Samotný webový portál není součástí projektu Eddie, kde bude záloha dat řešena jiným způsobem, proto jsem řešení záloh na webovém portálu v rámci této bakalářské práce vyřešil tímto polovičatým způsobem.

Záloha a obnovení databáze funguje pomocí T-SQL dotazů *BACKUP DATABASE* a *RES-*

TORÉ DATABASE respektive. Ovšem dříve, než je možné zálohu a obnovení databáze provést, je nutné se v rámci webové aplikace přepnout do jiné databáze a odhlásit všechny aktivní uživatele z databáze, kde chci zálohu provést. Soubor zálohy se ukládá na specifické místo na disku, odkud pak může webová aplikace získat přehled o všech provedených zálohách databází. K obnovení stačí ze seznamu vybrat konkrétní zálohu a webová aplikace nad ní zavolá zmíněný T-SQL dotaz.



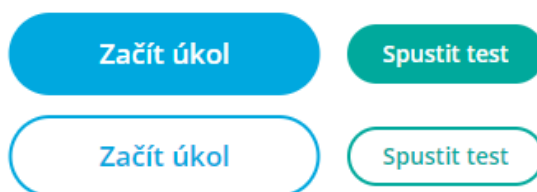
Obrázek 8: Uživatelské rozhraní zálohy a obnovení databáze

Při testování aplikace, když webový server nebyl spouštěn na stejném stroji jako databáze, jsem k vytváření záloh používal SQL Server Management Studio, které umožňuje ať už samotné provedení dříve zmíněných T-SQL dotazů, nebo také vygenerovat SQL skript k celé databázi.

7.2.3 Vzhled webové aplikace

Co se týče vzhledu, vystřídalala aplikace několik vzhledových variant. Nakonec o konečné podobě rozhodlo logo projektu Eddie, které se skládá ze čtyř základních barev. Rozhodl jsem se sladit web s těmito barvami, primárně se světle modrou a tyrkysovou.

K tvorbě kaskádových stylů jsem se rozhodl použít jazyk SASS, jelikož zkracuje dobu strávenou nad vytvářením kaskádových stylů. Knihovnu Bootstrap jsem použil pro tzv. normalizaci CSS — zajištění, že základní vzhled prvků webové aplikace se bude vykreslovat ve všech prohlížečích stejně. Jako druhá užitečná věc z této knihovny se ukázala responzivní mřížka, kterou jsem použil na formátování položek formulářů vyskytujících se v aplikaci.



Obrázek 9: Jednotná tlačítka napříč celým webem

Napříč webem jsem se rozhodl používat opakující se prvky — tlačítka mají zakulacené rohy a stejný efekt při najetí kurzorem. Panely v menu mají vždy stejnou HTML strukturu a vzhled. Při definování vzhledu těchto prvků jsem využil vlastnosti jazyka kaskádových stylů SASS, tzv. *mixin* — blok vlastností, který se v kódu nadefinuje jednou a poté jej lze vložit mezi vlastnosti jakéhokoli elementu. Tímto se vyhneme zbytečnému opakování kódu.

Webová aplikace je plně responzivní, zobrazí se bez problémů na kterémkoliv zařízení.

7.3 Návrh systému – diagnostický nástroj

Dospěl jsem k názoru, že test, který budou pacienti vykonávat, by měl být kontinuální, bez zbytečného přerušování. Proto jsem se rozhodl zaměřit se na asynchronní dotazování, které zajistí, že pro posun v testu nebude muset uživatel znovu načítat celou stránku. Asynchronní dotazování jsem se rozhodl implementovat pomocí technologie AJAX uvnitř JavaScriptu na straně klienta.

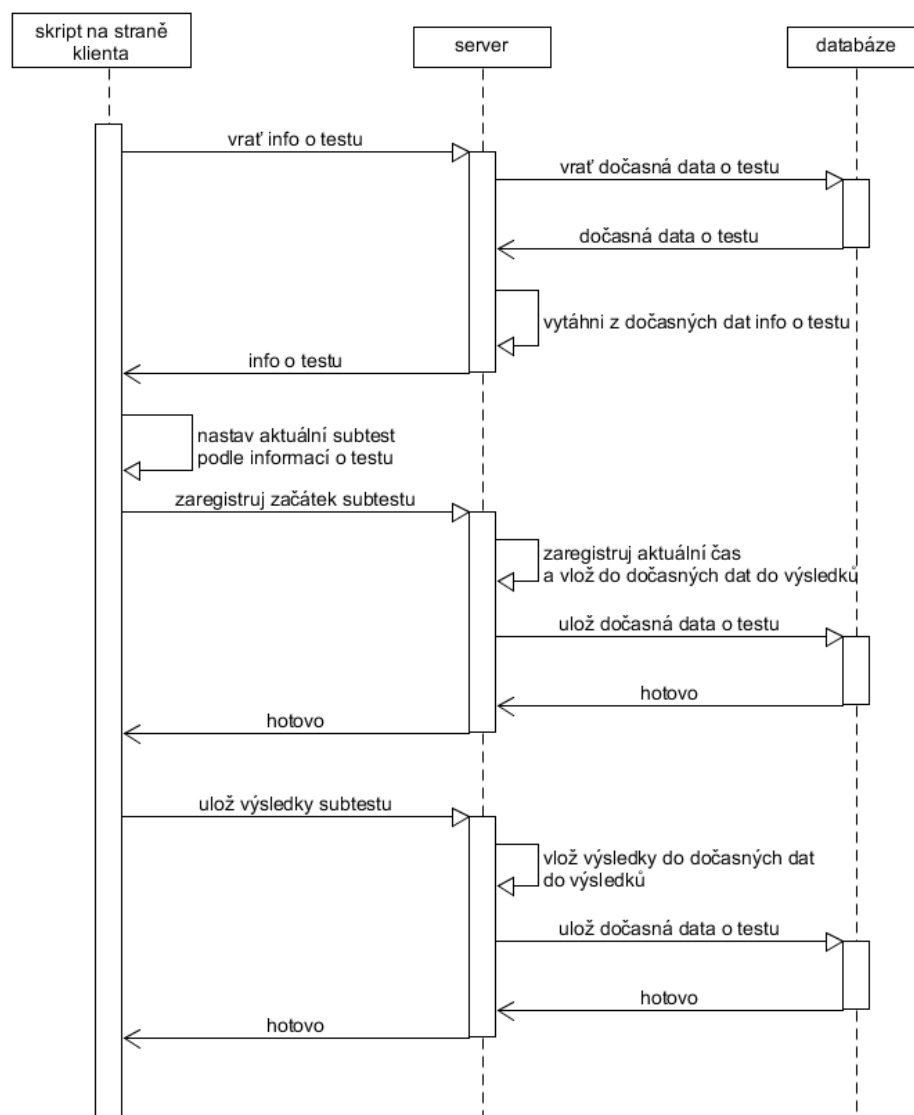
Dále jsem zvažoval možnost, že bych funkcionality subtestů, které fungují formou jednoduchých her, implementoval pomocí některého z JavaScriptových herních enginů, např. Phaser, Crafty nebo pixi.js. Nakonec jsem však zvolil pouze knihovnu jQuery, jelikož jsem usoudil, že vzhledem k jednoduchosti her v jednotlivých subtestech, by použití herního enginu bylo přehnané.

Když ovšem implementuji test pomocí JavaScriptu, nastane problém při vyhodnocování výsledků. Uživatel zkušený v oblasti webových technologií je schopen si v prohlížeči v developerské konzoli vypsát a hlavně upravit data, se kterými JavaScript aktuálně pracuje. Pokud bych zaznamenával délky vykonávání jednotlivých subtestů ze strany klienta, poskytl bych tím možnost zkušeným uživatelům upravit své výsledné časy ve svůj prospěch. Zmanipulovat tyto výsledky by šlo také změněním aktuálního času v nastavení operačního systému, jelikož JavaScript zjišťuje aktuální čas právě z času operačního systému stroje, na kterém je spouštěn.

Proto jsem musel navrhnout řešení, které by takovou manipulaci ze strany uživatele znemožnilo.

V momentě, kdy klient požaduje stránku, na které se nachází test, zkontroluje si webová aplikace, jestli v databázi neexistuje záznam dočasných testových dat, vázající se k uživateli. V případě, že neexistuje, vytvoří aplikace objekt dočasných dat, který se skládá z informací o testu a z průběžných výsledků. V informacích o testu se nachází ID aktuálního subtestu a ID verze testu.

Po přijetí odpovědi serveru se začne vykonávat JavaScript, fungující tak, jak je znázorněno na obrázku 10. V podstatě si pomocí asynchronního dotazu na server zjistí aktuální ID subtestu a ID verze testu, na základě těchto informací pak zobrazí aktuální subtest. Kdykoliv je třeba v testu změřit čas, volá skript asynchronně server vždy na začátku a na konci měření. Server si pak uloží čas začátku a čas konce měření, tyto časy spojí s aktuálním subtestem a data uloží jako dočasná testová data v binární podobě do databáze. Identicky to funguje i v případě ukládání jiných průběžných výsledků, například odpovědí, počtů kliknutí a jiných měřených hodnot.



Obrázek 10: Princip průběžného přeposílání výsledků na server

Dočasná testová data se ukládají do databáze z několika důvodů. Zaprvé proto, že aplikace na straně serveru funguje vždy v rámci jednoho HTTP požadavku a poté data, se kterými pracovala, zapomíná. Zadruhé to umožňuje uživateli vrátit se k testu, pokud omylem zavře okno prohlížeče, případně pokud nastane výpadek proudu. V databázi se rovněž ukládá čas poslední úpravy těchto dat. Pokud doba od poslední úpravy přetáhne jednu hodinu nebo-li pokud uživatel dlouhou dobu v testu nepokračuje, zobrazí se při následném HTTP požadavku test úplně od začátku.

Je třeba podotknout, že jsem tento systém měl až do doby testování aplikace vyřešen trochu jinak. Místo databáze jsem k ukládání těchto dočasných dat využíval relačních proměn-

ných. Po nasazení na slabší server se však vyskytl problém — IIS na hostingovém serveru začal z důvodu vyššího využití operační paměti opakovaně recyklovat aplikační pool. Při recyklaci aplikačního poolu se zahodí obsah relace a to zcela znemožňovalo běžné používání aplikace.

7.3.1 Obecná funkcionalita subtestů

Každý subtest je v rámci JavaScriptového kódu implementován jako třída *Task*, která v sobě obsahuje metody, které umožňují inicializaci subtestu, odeslání výsledků na server a po dokončení subtestu také vyčištění všech událostí, timeoutů a timerů, které subtest používá.

Výsledky každého subtestu jsou ukládány do JSON objektu, který je při dokončení subtestu odeslán pomocí AJAX na server, kde je spárován s odpovídajícím objektem dočasných dat pro daný subtest a uložen do databáze.

Každý subtest má specifickou strukturu HTML, která v sobě obsahuje dvě základní části — instrukce a samotný obsah úlohy. Tyto HTML soubory jsou uloženy jako šablony Razor ve složce Views ASP.NET aplikace. Před spuštěním subtestu vyvolá JavaScript asynchronní dotaz, který vrátí obsah tohoto HTML souboru a následně jej vloží do hlavního HTML elementu testu.

Při dokončení posledního subtestu se pomocí AJAX zavolá metoda *CompleteTest* na straně serveru, která spustí řetězec událostí vedoucí k zpracování dočasných testových výsledků a vypočtení finálních výsledků testu. Toto zpracování dočasných testových výsledků probíhá pomocí DLL knihovny diagnostického nástroje a obnáší vypočtení délky subtestů ze znalosti jednotlivých časových značek, vyhodnocení správnosti konkrétních odpovědí a další procesy. Zpracované finální výsledky testu jsou pak uloženy do databáze a skript je na straně klienta informován.

Asynchronní dotazy jsem zaobalil do funkce, která umožňuje hlavní skript formou návrhového vzoru Observer informovat o začátku a dokončení provádění dotazu. Při provádění jakéhokoliv AJAX dotazu se poté uživateli v aplikaci zobrazí ikonka načítání, která je zanimována pomocí kaskádových stylů.

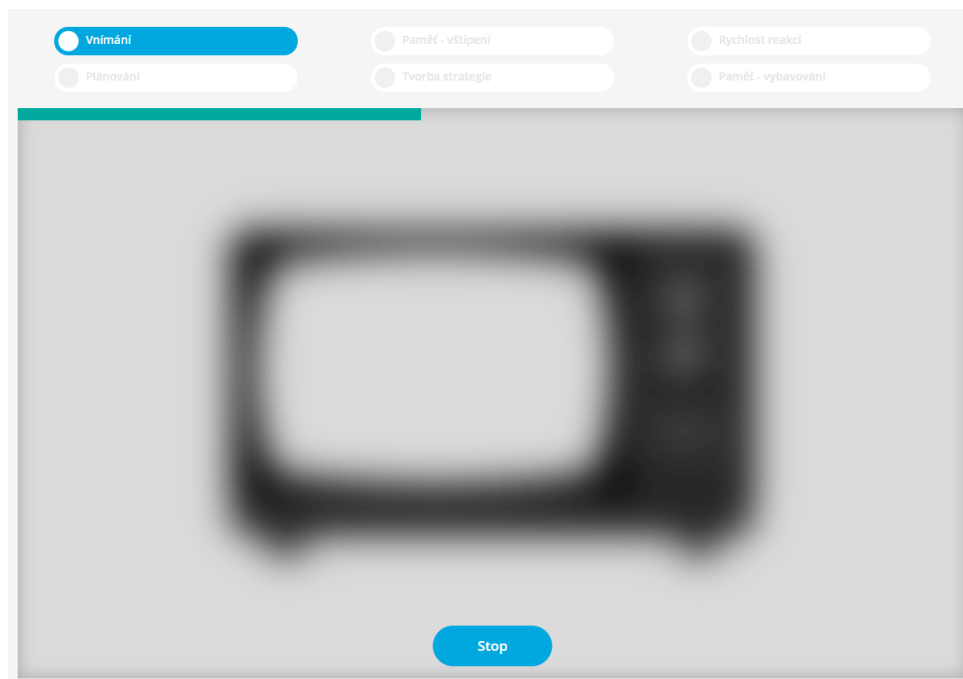
7.3.2 Dotazník CFQ

Při prvním spuštění testu se místo samotného testu zobrazí uživateli dotazník kognitivních chyb, který musí uživatel vyplnit. Dotazník se skládá z pětadvaceti otázek, na které proband odpovídá frekvencí, jak často se u něj dotazovaná věc děje. Odpovědi je pět s tím, že okrajové odpovědi jsou *nikdy* a *velmi často*. Tento dotazník je specifikován v rámci Neuropsychologické baterie. [43]

Dotazník jsem implementoval využitím klasického HTML formuláře generovaného pomocí ASP.NET.

7.3.3 Subtest – vnímání

Dle specifikace má úloha vnímání fungovat následovně — uživateli se zobrazí špatně rozpoznatelný objekt a úkolem bude zařadit tento objekt do kategorie, pod kterou patří.



Obrázek 11: Ukázka subtestu vnímání

Rozhodl jsem se využít CSS vlastnost *filter* a její parametr *blur*, který dokáže rozmazat HTML element, na který je tato vlastnost aplikována. Při zobrazení objektu pak nastavím hodnotu blur na určitý počet pixelů a postupně ji snižuji, tím se objekt postupně zaostruje.

```
task.blurElement = function () {
    var blurValue = this.progress.stepCount - this.progress.currentStep;
    this.$svgElement.css('filter', 'blur(' + blurValue + 'px)');
};
```

Výpis 6: JavaScriptová funkce upravující CSS vlastnost blur

Uživatel má k dispozici tlačítko, které po kliknutí zobrazí výběr ze čtyř kategorií: dopravní prostředek, rostlina, zvíře, přístroj.

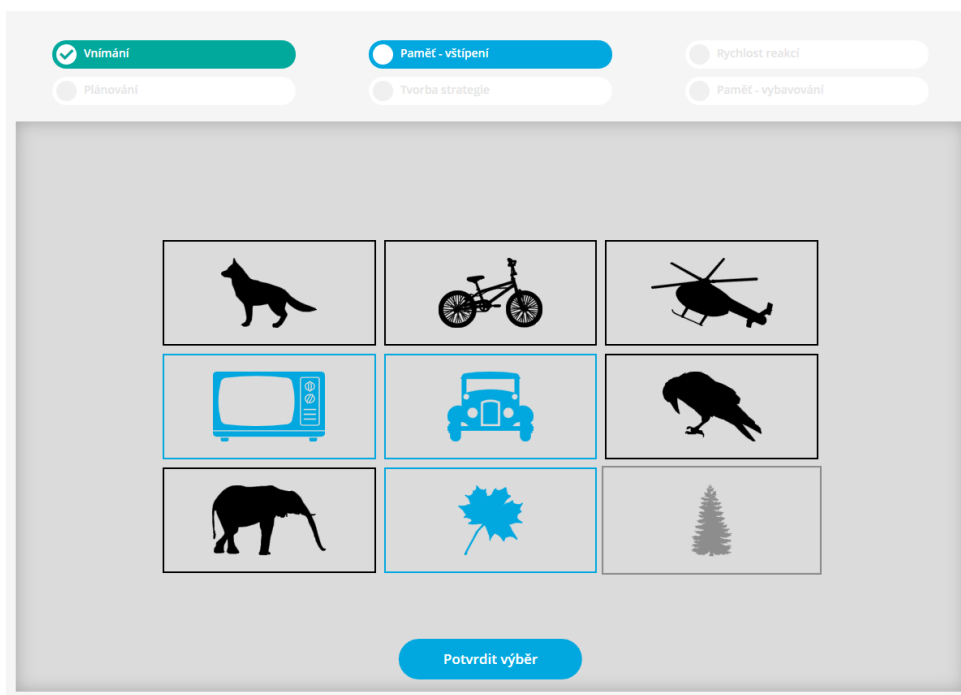
Objekty se zobrazují vždy ve stejném pořadí, ať už test vykonává kdokoli. Všechny objekty jsou černé siluety předmětů nebo zvířat. Zvolil jsem vektorovou grafiku (SVG), zvláště z důvodu lepší responzivity — obrázek lze donekonečna zvětšovat. SVG jsem získal z volně dostupných zdrojů, konkrétně se jedná o webovou stránku *pixabay.com*.

Měřené hodnoty jsou zde délka uplynulé doby od zobrazení objektu po kliknutí na tlačítko a také správnost odpovědi.

7.3.4 Subtest – pamětní proces vštípení

V tomto subtestu se uživateli zobrazí devět předmětů. Uživatel pak musí kliknutím označit ty předměty, které se vyskytly v předešlém úkolu.

Měří se zde uplynulá doba od momentu zobrazení této úlohy po potvrzení vybraných předmětů a také se zde určuje skóre dle následujících pravidel — za správně označené předměty se přičítá jeden bod a za špatně označené předměty se jeden bod odečítá.



Obrázek 12: Ukázka subtestu pamětního procesu vštípení

7.3.5 Subtest – rychlost reakcí

Tento subtest byl specifikován jako úkol, při kterém se uplatní uživatelská rychlost reakcí. Rozhodl jsem se zůstat u podobného vzhledu jako u předešlých úkolů a na náhodných pozicích zobrazovat uživateli obraz motýla (opět vektorová grafika). Uživatel pak má za úkol v průběhu jedné minuty odklikat co nejvíce motýlů.

```
task.$butterfly.show();
var x = Math.floor(Math.random() * task.columnsCount) * task.cellWidth;
var y = (Math.floor(Math.random() * task.rowsCount) * task.cellHeight) + task.
    safeZoneForHeight/2;
task.$butterfly.css({ 'top': y, 'left': x });
```

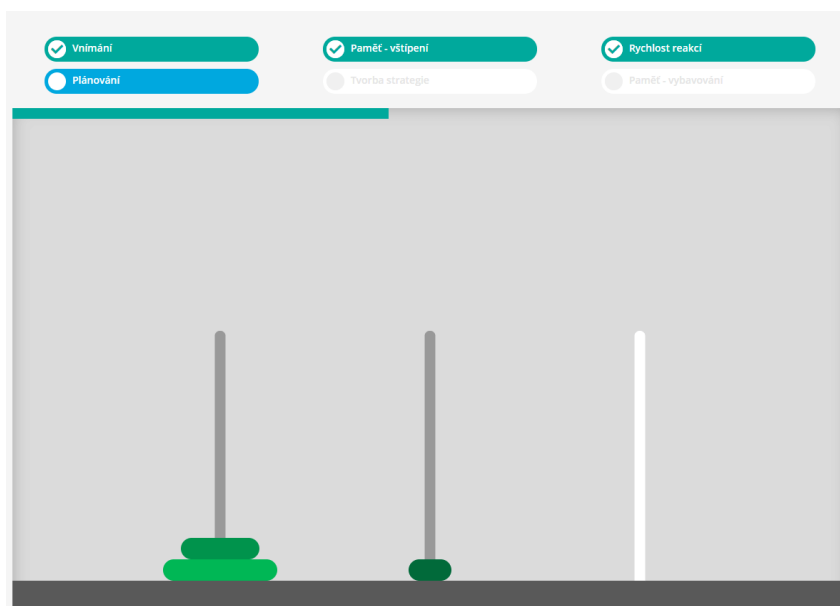
Výpis 7: JavaScriptová funkce na zobrazení motýla na náhodné pozici

Náhodnou pozici motýla skript určuje z mřížky, která má stanovený počet řádků a sloupců. Mřížka je responzivní, tudíž na menších displejích bude šířka sloupců a výška řádků nabývat menších hodnot. Pozice motýla je pak náhodnou buňkou v této mřížce, viz výpis 7.

Zde se měří počet odklikaných motýlů a také počet, kolikrát uživatel klikl mimo motýla.

7.3.6 Subtest – plánování

Subtest zaměřený na proces plánování obsahuje klasické Hanojské věže. Jednotlivé kotouče lze přemísťovat kliknutím levým tlačítkem myši.



Obrázek 13: Ukázka subtestu zaměřeného na proces plánování

Uživatel musí vyřešit pod tlakem časového limitu tři kola Hanojských věží, všechna tři kola stejné obtížnosti. V prvních verzích se obtížnost navyšovala, nakonec po několika konzultacích jsem spolu s dalšími lidmi došel k závěru, že bude lepší obtížnost nechat vždy stejnou. Někteří lidé mají problémy již s třemi kotouči, nemluvě o tom, jaké problémy mohou mít při řešení této úlohy lidé s poškozením mozku.

Ve skriptu jsou jednotlivé věže a kotouče reprezentovány objektovým kódem. Při změně je zavolána funkce *updateView*, která zaktualizuje grafický výstup, viz výpis 8.

```
task.updateView = function () {
  $.each(this.towers, function (k, tower) {
    $('<div>.tower[data-id=' + k + '</div>').empty();

    if (tower.towerStack.length > 0) {
      var towerId = k;
```

```

        $.each(tower.towerStack, function (k, towerPiece) {
            var towerPieceHtml = '<span class="tower-piece tower-piece-size-' +
                towerPiece.size + '" style="bottom: ' + k * 30 + 'px;"></span>';
            $('<div>'.tower[data-id="' + towerId + '"]').append(towerPieceHtml);
        });
    }
});

if (this.movingTowerPiece != null) {
    this.$movingTowerPiece.show();
    this.$movingTowerPiece.removeClass('tower-piece-size-1 tower-piece-size-2
        tower-piece-size-3 tower-piece-size-4 tower-piece-size-5');
    this.$movingTowerPiece.addClass('tower-piece-size-' + this.
        movingTowerPiece.size);
}
else {
    this.$movingTowerPiece.hide();
}
};

```

Výpis 8: JavaScriptová funkce na aktualizování grafického výstupu Hanojských věží

Kromě délky řešení jednotlivých kol se v této úloze měří také počet tahů a počet špatných tahů — tedy tahů, kdy se uživatel pokusil položit větší kotouč na menší.

7.3.7 Subtest – tvorba strategie

V úloze testující schopnost vytvářet strategie byly na základě konzultací zvoleny piškvorky v poli o třech sloupcích a třech řádcích.

Počítačového protivníka jsem se rozhodl implementovat pomocí algoritmu *minimax* [44]. Tento algoritmus je využíván pro AI u tahových her, třeba právě u piškvorek nebo u šachů.

Princip algoritmu minimax spočívá v možnosti počítače vidět několik tahů dopředu. Počítač je schopen pomocí rekurze nasimulovat každý možný tah, který může nastat v konkrétním hracím poli. Jakmile se v simulaci dostane do konečného stavu (výhra, prohra, remíza), ohodnotí tento stav kladným počtem bodů při výhře, záporným počtem bodů při prohře a žádnými body při remíze.

Následně propaguje tu větší hodnotu ze svých tahů a menší hodnotu ze simulovaných tahů hráče. Tedy maximalizuje v případě tahu AI, minimalizuje v případě tahu hráče — odtud název minimax. Pak počítač prochází ohodnocené tahy a vybere si ten, který je ohodnocen největším počtem bodů.

Tento algoritmus způsobuje také neporazitelnost počítače. Proto cílem piškvorek v tomto subtestu není vyhrát, ale zamezit počítači, aby vyhrál.

V předešlých verzích této aplikace jsem implementoval také piškvorky ve větších polích. Už ve verzi v poli o čtyř sloupcích a čtyř řádcích však nastal problém — tah počítače trval příliš dlouhou dobu. Je to způsobeno tím, že při procházení všech možných tahů se potýkáme s číslem $16!$, tedy $2.1 \cdot 10^{13}$ na rozdíl od čísla $9!$ ve verzi piškvorek v poli o třech sloupcích a třech řádcích.

Když počítač simuluje všechny možné tahy, simuluje velké množství tahů, které jsou v podstatě identické s předešlými, zbytečně tedy takto prodlužuje čas k vykonání algoritmu. Naštěstí existuje metoda, která zajistí, že počítač tyto redundantní tahy započítávat nebude. Tato metoda se nazývá *alfa-beta ořezávání*.

Principem alfa-beta ořezávání je, že pokud již byla do simulovaného tahu počítače propagována nejvyšší možná hodnota, nemusí se simulovat další kombinace vycházející z tohoto tahu. Totéž platí v simulovaném tahu hráče, pokud už byla propagována nejmenší možná hodnota.

```
task.minimax = function (newGrid, player, alfa, beta) {
  var emptyCells = task.getEmptyCells(newGrid);
  var move = {};
  var nextMoveIndex;

  if (task.hasWon(newGrid, task.player)) {
    return { score: -10 };
  }
  else if (task.hasWon(newGrid, task.ai)) {
    return { score: 10 };
  }
  else if (emptyCells.length === 0) {
    return { score: 0 };
  }

  for (var i = 0; i < emptyCells.length; i++) {
    move.index = newGrid[emptyCells[i]];
    newGrid[emptyCells[i]] = player;

    if (player === task.ai) {
      var result = task.minimax(newGrid, task.player, alfa, beta);

      if (result.score > alfa) {
        alfa = result.score;
        nextMoveIndex = move.index;
      }
    }
    else {
      var result = task.minimax(newGrid, task.ai, alfa, beta);

      if (result.score < beta) {
        beta = result.score;
        nextMoveIndex = move.index;
      }
    }

    newGrid[emptyCells[i]] = null;
  }

  return { score: nextMoveIndex ? result.score : 0 };
}
```

```

    }
  }
  else {
    var result = task.minimax(newGrid, task.ai, alfa, beta);

    if (result.score < beta) {
      beta = result.score;
      nextMoveIndex = move.index;
    }
  }

  newGrid[emptyCells[i]] = move.index;

  if (alfa >= beta) {
    break;
  }
}

return { index: nextMoveIndex, score: (player === task.ai) ? alfa : beta };
};

```

Výpis 9: Minimax s alfa-beta ořezáváním implementovaný v JavaScriptové funkci

7.3.8 Subtest – pamětní proces vybavování

Jako poslední subtest jsem pouze zduplikoval úlohu ze subtestu zaměřeného na pamětní proces vštípení. Tento subtest spoléhá na to, že proběhla dostatečně dlouhá doba od prvního úkolu a lidé budou mít větší problém vybavit si předměty, které se tam vyskytly.

7.3.9 Vzhled testu

Vzhled testu jsem navrhl tak, aby odpovídal vzhledu webového portálu. Všechny subtesty tedy zapadají do barevného schématu portálu včetně tlačítek, které jsou identické s těmi, které se vyskytují na jiných částech webu.

Samotná stránka, kde se test vyskytuje, se automaticky nastavuje na výšku viewportu — okna, ve kterém je vykreslován obsah webové stránky v prohlížeči. Obsahuje také zjednodušenou patičku a hlavička se automaticky skrývá. Tyto věci jsou důležité z důvodu eliminace rušivých faktorů při vykonávání testu.

V rámci jednotlivých subtestů používám ikonky z webového toolkitu Font Awesome, který v době psaní tohoto textu poskytuje zhruba 1500 vektorových ikon k použití zdarma.

Responzivita testu je funkční jen po šířku viewportu notebooků. Na menších zařízeních nemělo smysl řešit zobrazování subtestů, jelikož by test v této velikosti přinášel mnoho problémů. Problémy s hodnocením testových úloh — například v subtestu rychlosti reakcí by lidé dosahovali na menších zařízeních lepších výkonů z důvodu menších vzdáleností mezi motýli; starší lidé by naopak mohli mít problém rozeznat malé tvary na obrazovce.

7.3.10 Vyhodnocování testu

Logika k vyhodnocování výsledků včetně již zmíněné administrace testu je implementována jako DLL knihovna. Rozhodl jsem se tak zpočátku z důvodu větší modularity testovacího nástroje. Později, když byl dohodnut jiný postup napojení testu do projektu Eddie, jsem tuto strukturu ponechal, zvláště z důvodu rozsáhlosti tohoto modulu.

Cílem vyhodnocení je určit skóre v rámci *CHC modelu inteligence* [45] a předat uživateli informaci, jaký je jeho výkon v jednotlivých oblastech CHC modelu. Mezi tyto oblasti patří kvantitativní znalosti, čtení a psaní, schopnost porozumět, přizpůsobivé myšlení, krátkodobá paměť, dlouhodobá paměť a vybavování, zrakové procesy, sluchové procesy a rychlost zpracování informací.

Některé tyto oblasti nejsou pokryty testem, nýbrž CFQ dotazníkem, který uživatel musí vyplnit před samotným testem.

Klíčové tedy bylo vymyslet postup, jak se z množiny hodnot parametrů měřených při provádění testu dostat na skóre jednotlivých oblastí CHC modelu. První bylo třeba ohodnotit každý měřený parametr v testu a položku v CFQ dotazníku 0 – 4 body pro každou oblast, s tím že 0 bodů znamená, že tato oblast nemá vůbec žádný vliv na měřený parametr a 4 body znamenají, že tato oblast má maximální vliv na měřenou parametr. Toto ohodnocení bylo provedeno dvěma psychology.

Následně bylo třeba navrhnout dynamickou normu. Ta byla navržena tak, že při dokončení testu se vypočtou průměrné, minimální a maximální hodnoty všech hodnot měřených parametrů ze všech testů uživatelů, kteří spadají pod stejnou skupinu co se týče věku, pohlaví a vzdělání. Výsledek se zapíše do databáze do tabulky norem.

Posledním požadavkem bylo zajistit, že aplikace ohodnotí kratší čas větším skóre než delší čas, větší počet správně uhodnutých objektů větším počtem bodů atd. Toto bylo zajištěno pomocí pole měřených položek, ve kterém se mimo již zmíněného hodnocení parametrů v rámci oblastí CHC modelu nachází také speciální parametr, která určuje, zdali má být menší hodnota měřeného parametru v testu hodnocena větším počtem bodů nebo naopak.

Výsledné skóre v rámci oblastí CHC modelu se poté vypočte následujícím způsobem:

1. Procházím postupně všechny parametry testu a CFQ dotazníku.
2. Pro každý parametr si zjistím jeho maximum, minimum a průměr v rámci stejné skupiny uživatelů.

3. Otestuji, jestli se hodnota parametru nachází nad průměrem, nebo pod průměrem.
4. Vypočtu $\frac{|h - \emptyset|}{M - \emptyset}$ kde h odpovídá hodnotě parametru, \emptyset aritmetickému průměru a M maximu nebo minimu.
5. Výsledek z minulého bodu vynásobím mínus jednou, pokud větší hodnota u tohoto parametru má být hodnocena nižším skóre.
6. Vzniklé procento v desetinném tvaru vynásobím počtem bodů u jednotlivých oblastí z CHC modelu. Vznikne mi skóre parametru ve všech oblastech CHC modelu.
7. Po projití všech parametrů se ke skóre u každé oblasti CHC modelu v rámci celkového testu a CFQ dotazníku dostanu zprůměrováním všech skóre z jednotlivých parametrů.

Tento postup výpočtu skóre byl odsouhlasen při konzultacích s odborníkem z klinické praxe.

Následně se uživateli u každé oblasti CHC modelu zobrazí, zdali je zde podprůměrný, průměrný nebo nadprůměrný. Toto je vyvozeno ze skóre v oblastech CHC modelu, které je menší než -0.1, mezi -0.1 a 0.1 nebo větší než 0.1, respektive.

Jak vyhodnocení vypadá v uživatelském rozhraní, je možno nalézt v příloze A.

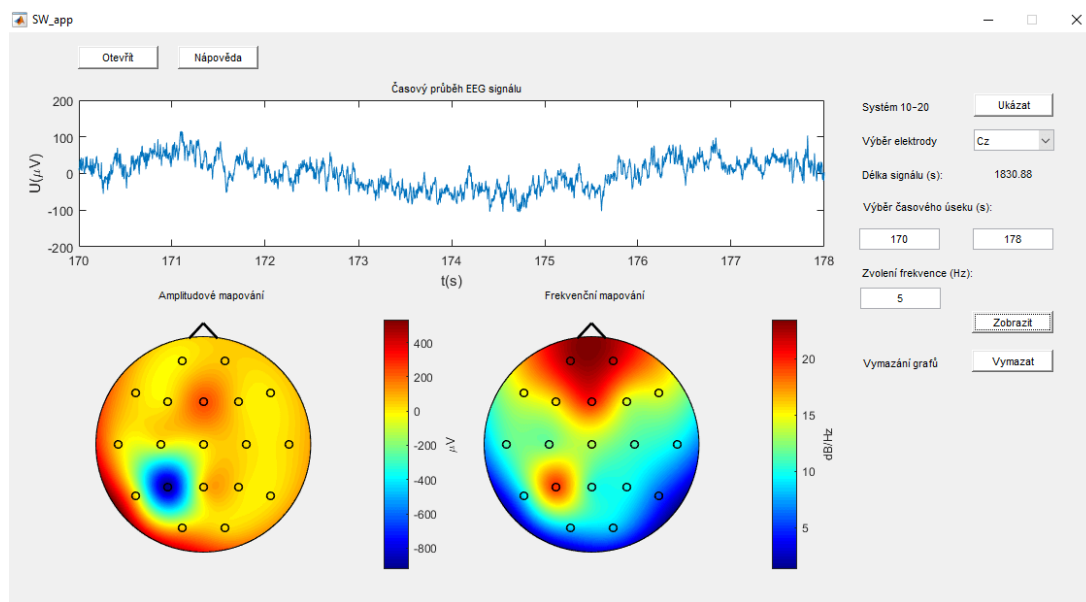
8 Ověření funkčnosti navrženého řešení

Výsledek práce jsem rozdělil do několika částí. V první části rozebírám výsledky nasazení a následného používání webové aplikace. Dále rozebírám výsledky testování na stovkách uživatelích spolu se základním statistickým zpracováním těchto výsledků. Nakonec se zaměřuji na zhodnocení souvstažnosti dotazníku s testem a popisuji, jak je samotná webová aplikace schopna tyto výsledky testů zanalyzovat.

8.1 Ověření funkčnosti

Webový portál včetně databáze byl nasazen na cloudovou službu Amazon Web Services. Odkaz na webovou stránku byl rozeslán pomocí sociální sítě Facebook. Na webovou aplikaci se za dobu zhruba dvou týdnů registrovalo 208 uživatelů. Test obou verzí a dotazník vyplnilo celkově 106 lidí, i přes to, že požadovaným počtem probandů v zadání bakalářské práce bylo probandů 30.

Mezi zbývajících 102 uživatelů patří lidé, kteří se registrovali, ale měli problém s potvrzením účtu skrze e-mail (e-mail jim nedošel, případně přišel do složky spamu nebo trvalo delší dobu, než přišel). Patří tam také lidé, kteří si účet aktivovali, ale nevyplnili dotazník; lidé, kteří vyplnili dotazník ale z aktuálně neznámých technických důvodů nemohli pokračovat ve vykonávání testu; případně největší skupina lidí, kteří neprovedli test verze B — tento počet lidí by pravděpodobně byl značně vyšší, kdybych v průběhu testování do aplikace nepřidal zprávu po vykonání testu verze A, která uživateli oznámí, že má provést i verzi B.



Obrázek 14: Grafické zobrazení mozkové aktivity člověka při vykonávání subtestu s Hanojskými věžemi [46]

Zároveň jsem aplikaci včetně databáze nasazoval také na český ASP.NET hosting, kde jsem narazil na problém s vysokým využitím operační paměti, který způsobil recyklování aplikačního

poolu. Jak jsem se již dříve v této práci zmínil, recyklováním aplikačního poolu se ze serveru ztrácel obsah relací, což představovalo problém v dřívější funkcionalitě skrze relační proměnné. Tento problém jsem vyřešil skrze užití cookie souborů a ukládáním dočasných výsledků testů do databáze.

Aplikaci jsem měl také možnost otestovat na člověku, kterému byla měřena mozková aktivita pomocí metody brain mappingu. K tomuto testu došlo díky spolupráci se studentkou biomedicínského inženýrství Evou Malaníkovou.

Na obrázku 14 je k nahlédnutí graf mozkové aktivity v momentě, kdy testovaný prováděl subtest zaměřený na plánování (Hanojské věže). Je zde aktivní hlavně prefrontální oblast, která se vyznačuje zvýšenou aktivitou při řešení problémů, rozhodování a plánování. [46]

8.2 Statistické zpracování měřených dat

Výsledky testu se skládají z osmatřiceti parametrů. Jedná se o časy jednotlivých subtestů, čas celkového testu, správnosti řešení, počty kliknutí a jiné (viz popisy jednotlivých subtestů v předešlé kapitole).

Pro názornost byly vybrány čtyři parametry a u každého z nich znázorněny výsledky všech probandů vhodnou formou jednosměrné analýzy.

U každého měřeného parametru bylo naměřeno 121 hodnot ze všech testů verze A. Vzorek populace, který byl měřen, jsou muži a ženy ve věku 20 – 65 let, minimálně s dokončeným středoškolským vzděláním.

8.2.1 Počet správně uhodnutých figur v subtestu vnímání

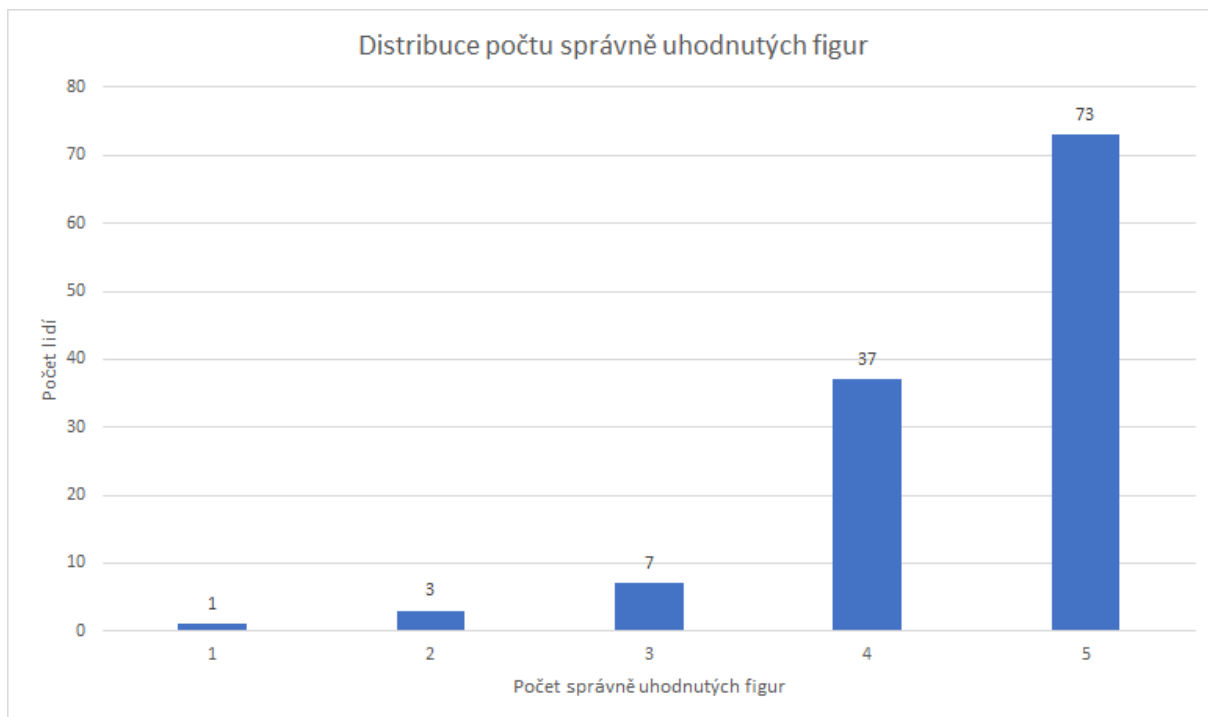
Jak již název napovídá, tento parametr znázorňuje, kolik figur uživatel zařadil do správných kategorií v subtestu vnímání. Hodnota parametru vzniká porovnáním se správnými výsledky na straně serveru při zpracování dočasných testových výsledků. Tento proces zpracování dočasných testových výsledků probíhá po dokončení celého testu.

V tomto subtestu se zobrazuje pět různých figur, tudíž největší počet správně uhodnutých figur je roven právě pěti.

Podle bodového hodnocení jednotlivých oblastí CHC modelu dvěma psychology, je výsledek hodnoty tohoto parametru testu ovlivněn zvláště zrakovými procesy (visual processing) a přízpůsobivým myšlením (fluid reasoning).

Na obrázku 15 se nachází graf distribuce tohoto parametru testu a v tabulce 1 se nachází jeho aritmetický průměr, střední hodnota a směrodatná odchylka. Graf i statistické veličiny byly vypočteny ze všech výsledků testu.

Lidé nejčastěji správně uhodli všech pět figur (73 lidí), případně udělali jednu chybu (37 lidí).



Obrázek 15: Graf distribuce počtu správně uhodnutých figur

Veličina	Hodnota
Aritmetický průměr	4.47
Střední hodnota	5
Směrodatná odchylka	0.79

Tabulka 1: Statistické veličiny vypočtené z počtu správně uhodnutých figur

8.2.2 Počet bodů v subtestu pamětního procesu vštípení

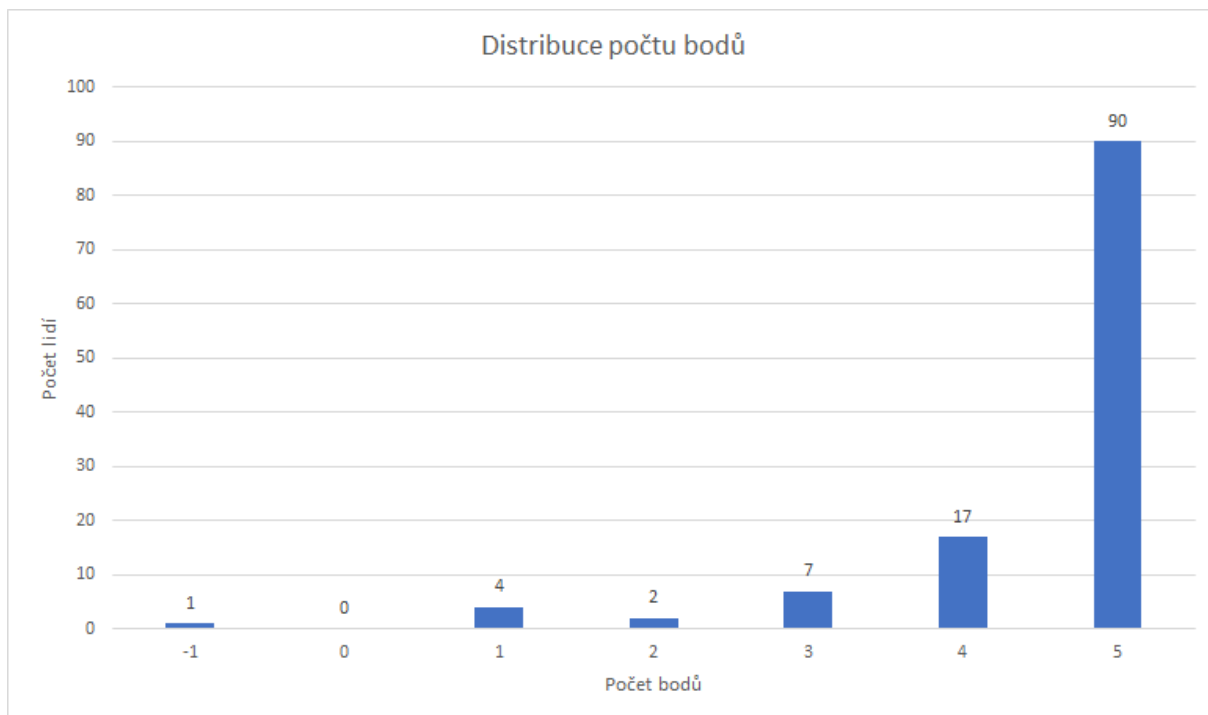
Počet bodů v tomto subtestu se počítá podle počtu správně označených figur, tedy figur, které se vyskytly v subtestu vnímání.

Výpočet se řídí těmito pravidly: za správně označené figury se body přičítají a za špatně označené figury se body odečítají. Nejmenší možný počet bodů je -4 a největší možný počet bodů je 5.

Podle bodového hodnocení jednotlivých oblastí CHC modelu dvěma psychology, je výsledek hodnoty tohoto parametru testu ovlivněn zvláště krátkodobou pamětí (short-term memory).

Opět zde znázorňuji výsledky tohoto parametru testu pomocí sloupcového grafu a statistických veličin v tabulce 2.

Z grafu je možné vyčíst, že většina lidí, kteří prováděli tento subtest, nemělo problém správně označit všechny figury vyskytující se v první úloze.



Obrázek 16: Graf distribuce počtu bodů

Veličina	Hodnota
Aritmetický průměr	4.51
Střední hodnota	5
Směrodatná odchylka	1.06

Tabulka 2: Statistické veličiny vypočtené z počtu bodů ze subtestu pamětního procesu vstípení

8.2.3 Celkový počet tahů v subtestu zaměřeném na proces plánování

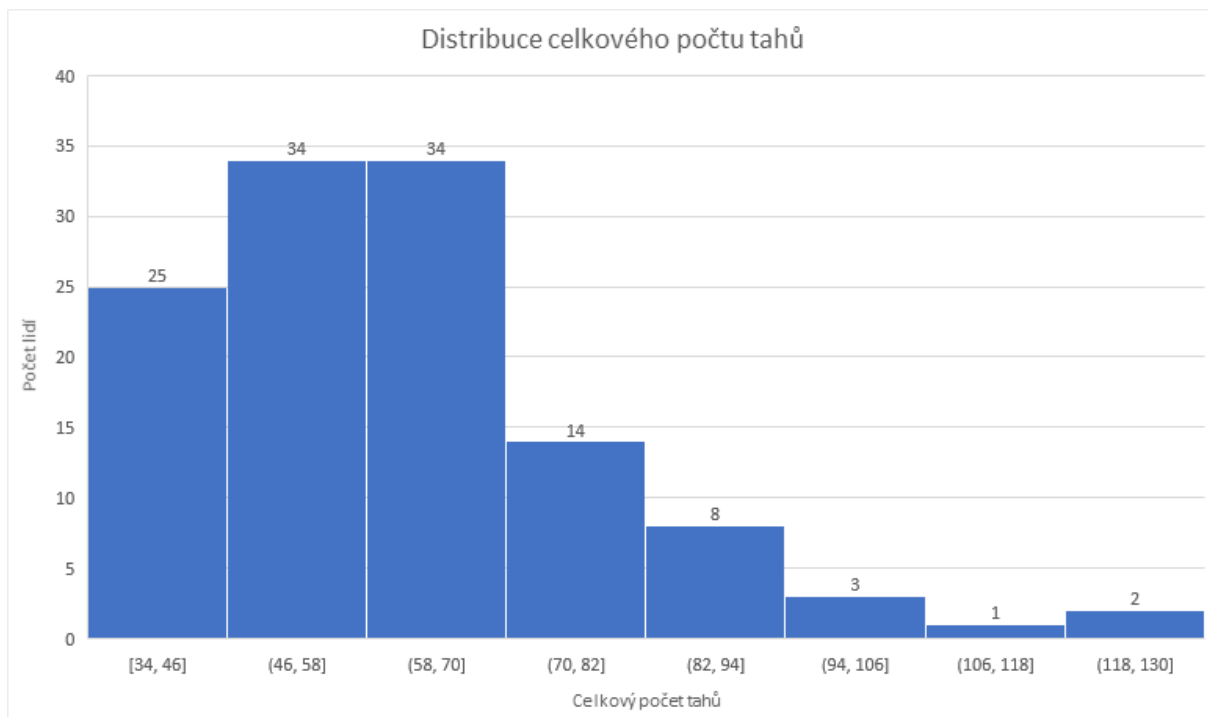
V tomto subtestu hraje uživatel Hanojské věže. Celkový počet tahů se získá sečtením tahů ze všech tří kol. Toto sečtení opět probíhá až po dokončení testu. Tento subtest je omezen časovým limitem, konkrétně jednou minutou v každém kole.

Jako jeden tah se počítá zvednutí kotouče nebo položení kotouče. Do celkového počtu tahů se počítají i špatné tahy (pokus o položení většího kotouče na menší).

Podle bodového hodnocení jednotlivých oblastí CHC modelu dvěma psychology, je výsledek hodnoty tohoto parametru testu ovlivněn zvláště rychlostí zpracování informací (processing speed), zrakovými procesy (visual processing) a přizpůsobivým myšlením (fluid reasoning).

Výsledky jsou znázorněny pomocí statistických veličin v tabulce 3. U tohoto parametru distribuci výsledků znázorňují z důvodu většího počtu různých hodnot pomocí histogramu na obrázku 17.

Nejčastější celkový počet tahů se dle výsledků pohybuje mezi 46 a 70 tahy.



Obrázek 17: Graf distribuce celkového počtu tahů

Veličina	Hodnota
Aritmetický průměr	62.03
Střední hodnota	59
Směrodatná odchylka	17.10

Tabulka 3: Statistické veličiny vypočtené z celkového počtu tahů

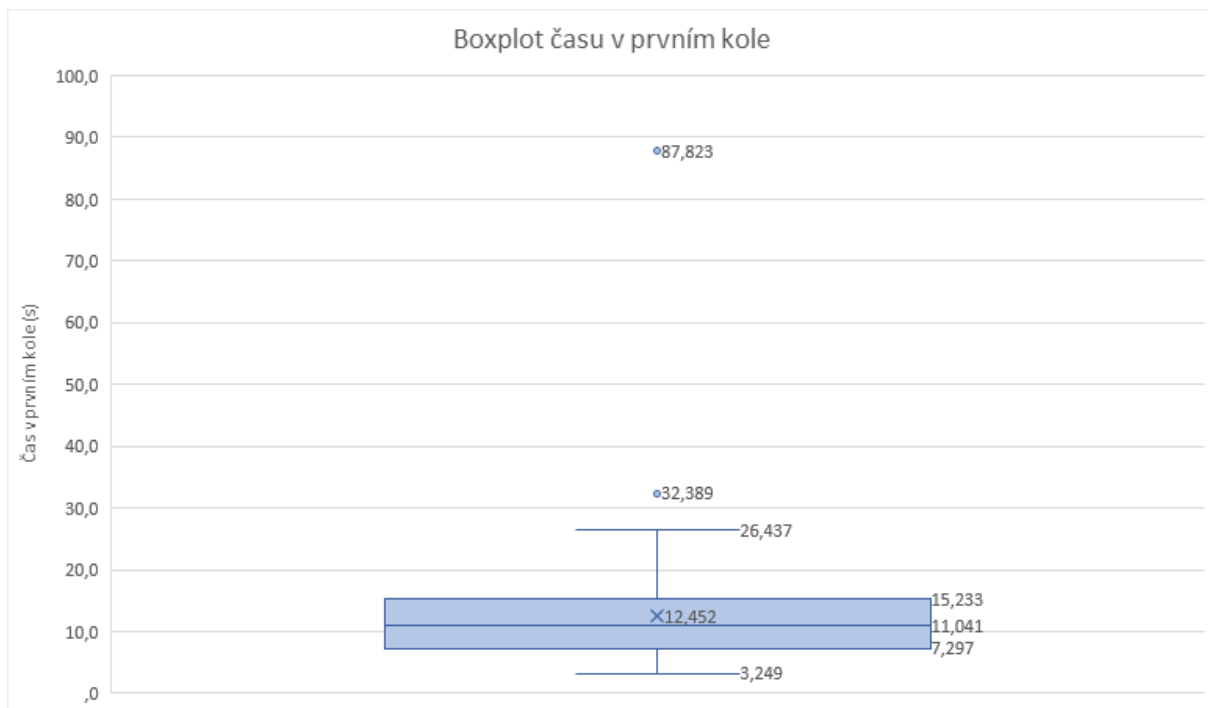
8.2.4 Čas prvního kola v subtestu zaměřeném na tvorbu strategie

V tomto subtestu hraje uživatel piškvorky s počítačem. Čas prvního kola se vypočte při dokončení celého testu z časových značek vytvořených na straně serveru při začátku a konce prvního kola.

Podle bodového hodnocení jednotlivých oblastí CHC modelu dvěma psychology, je výsledek hodnoty tohoto parametru testu ovlivněn zvláště kvantitativními znalostmi (quantitative knowledge) a přizpůsobivým myšlením (fluid reasoning).

Zde znázorňuji výsledky kromě užití statistických veličin také pomocí krabicového grafu vzhledem k tomu, že data nejsou kategorická.

Z grafu lze mimo jiné zjistit, jaký je zde dolní a horní kvartil — 7.3 a 15.2 sekund.



Obrázek 18: Krabicový graf času prvního kola

Veličina	Hodnota
Aritmetický průměr	12.45
Střední hodnota	11.04
Směrodatná odchylka	9.07

Tabulka 4: Statistické veličiny vypočtené z času prvního kola (v sekundách)

8.3 Zhodnocení souvztažnosti dotazníku CFQ s testem

K určení souvztažnosti dotazníku s testem byla využita korelace skóre oblastí CHC modelu z výsledků CFQ dotazníku se skóre oblastí CHC modelu z výsledků testu.

Tímto je možno definovat, jestli skóre vypočtené z výsledků CFQ dotazníku poměrově odpovídá skóre vypočteného z výsledků testů.

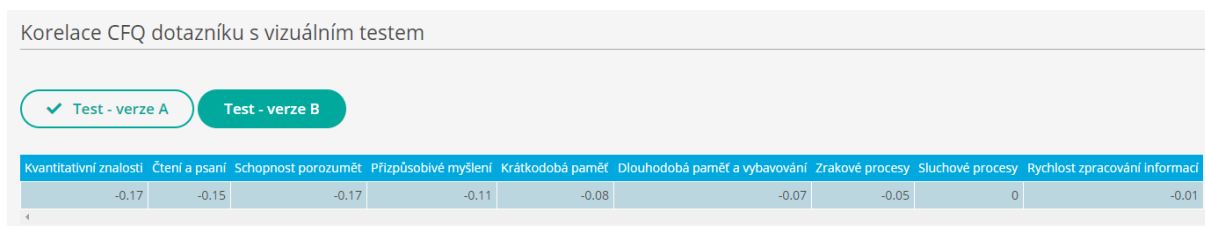
Korelace je určena pomocí Pearsonova korelačního koeficientu [47], který se vypočte následovně:

$$r = \frac{n \sum_{i=1}^n (x_i y_i) - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{(n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2)(n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2)}$$

kde x a y jsou skóre a n je počet dvojic skóre.

Tento výpočet byl implementován v rámci webového portálu, kde je umožněno správcům zobrazit si korelaci u každé oblastí CHC modelu.

Výsledné korelace, které vznikly z výsledků všech 121 testů verze A a vyplněných dotazníků, jsou znázorněny na obrázku níže.



Obrázek 19: Korelace CFQ dotazníku a testu

Výsledné korelační koeficienty se pohybují blízko nule, což značí, že korelace mezi skóre z CFQ dotazníku a testem je minimální až neexistující. Navíc jsou tyto korelační koeficienty záporné. To znamená, že čím větší je skóre z CFQ dotazníku, tím naopak menší je skóre vypočtené z testu. V oblasti sluchové procesy je korelace nulová, z toho důvodu, že se test na tuto oblast nezaměřoval.

Důvod výsledné korelace může být v tom, že hodnocení jednotlivých parametrů testu bylo vytvářeno pouze ze zkušeností a odhadu dvou psychologů, nejedná se o výsledek měření. Vliv může mít také pravdivost odpovědí v CFQ dotazníku a faktory, které mohou ovlivňovat výsledek testu — prostředí, ve kterém člověk test provádí, technika, kterou k testu člověk používá (počítačová myš vs. touchpad) a také schopnosti testovaného člověka ovládat počítač.

Byly vybrány dvě oblasti CHC modelu, konkrétně kvantitativní znalosti a zrakové procesy, u kterých byl vypracován graf, který znázorňuje tyto korelace. Tyto grafy jsou k dispozici v příloze B.

8.4 Vyhodnocení měřených dat ve webové aplikaci

Správce má ve webové aplikaci možnost zobrazit si aktuální výsledky formou grafu. Graf zobrazuje distribuci každé měřené položky, a to formou sloupcového grafu v případě, že se jedná o kategorická data a nebo formou histogramu, v případě, že se jedná o numerická data.

Grafy (viz příloha C) jsou vykreslovány pomocí JavaScriptové knihovny Plotly. Je zde možnost výsledky testů filtrovat a to dle verze testu, věkové kategorie, dosaženého vzdělání a pohlaví testovaného.

Pod grafem se rovněž zobrazuje počet filtrovaných výsledků testu, minimální hodnota, maximální hodnota, střední hodnota, aritmetický průměr a směrodatná odchylka.

V průběhu řešení bakalářské práce vznikl po konzultacích s odborníkem z klinické praxe požadavek na určení reliability testu. Ukázalo se však, že se jedná o časově náročnou úlohu, která je nad rámec předkládané práce a není možné ji v rámci této práce do webové aplikace implementovat. Tento fakt však nevylučuje možnost dalšího rozšíření této práce například o implementaci této funkcionality.

9 Závěr

Výsledkem předkládané práce je webový portál obsahující diagnostický nástroj pro hodnocení kognitivních funkcí. Webový portál zajišťuje správu uživatelských účtů a zabezpečení dat realizované hashováním hesla s použitím soli a užitím pouze jednoho osobního údaje k propojení dat s konkrétním člověkem — e-mailové adresy, která je ukládána v hashované podobě. Webový portál taktéž splňuje zásady obecného nařízení o ochraně osobních údajů — uživatelé musí při registraci udělit souhlas s využitím svých osobních údajů pro účely testování a také mají k dispozici možnost kdykoliv svůj uživatelský účet odstranit, případně anonymizovat.

Diagnostický nástroj se skládá z dotazníku kognitivních chyb a interaktivního testu ve dvou paralelních verzích, realizovaného formou jednoduchých her a zaměřujícího se na jednotlivé kognitivní procesy, kterými jsou vizuální zpracovávání (vnímání), fungování pracovní paměti ve vizuální paměti, pozdější vybavení naučeného, exekutivní funkce (schopnost plánování a udržení niterného plánu) a celkové psychomotorické tempo.

Výsledky testů jsou zpracovávány formou jednosměrné analýzy, skóre je počítáno pro každou oblast CHC modelu inteligence, dynamicky je zpracovávána souvztažnost CFQ dotazníku s testem, k jejíž určení je využita korelace skóre oblastí CHC modelu inteligence z výsledků CFQ dotazníku se skóre oblastí CHC modelu inteligence z výsledků testu.

Jednotlivé úlohy testu byly navrženy tak, aby uživatelům nepředstavovalo problém je vykonat. Vzhled testu je sladěn se vzhledem webového portálu, je zde kladen důraz na jednoduchost a minimální výskyt rušivých elementů. Webový portál je plně responzivní, samotný test však nelze zobrazit na zařízeních menších, než notebook — zde by nastal problém s čitelností některých částí testu a některé úlohy by se pravděpodobně zjednodušily.

Zmíněný diagnostický nástroj byl nasazen na dva různé servery a otestován na 208 uživatelích a také na jednom probandovi při měření mozkové aktivity pomocí metody EEG brain mappingu. Z celkového počtu uživatelů registrovaných do webového portálu vykonalo obě paralelní verze testu 106 lidí, zbytek uživatelů vykonalo pouze jednu verzi testu, případně jen dotazník CFQ.

Výsledek testů byl znázorněn na čtyř vybraných parametrech testu formou jednosměrné analýzy a určením aritmetického průměru, střední hodnoty a směrodatné odchylky těchto parametrů. Byla prokázána malá až neexistující souvztažnost CFQ dotazníku s testem. Důvodem tohoto výsledku by mohlo být odhadem určené hodnocení měřených parametrů testu, které bylo provedeno dvěma psychology, nebo také pravdivost odpovědí v CFQ dotazníku a vliv různých faktorů na provádění testu — různorodost zařízení, na kterém test může být vykonán, úroveň schopností uživatele s tímto zařízením pracovat a prostředí, ve kterém uživatel test prováděl.

Validitu diagnostického nástroje bude nutné ověřit formou srovnání se stávajícími psychologickými testy na reprezentativním populačním souboru jak neklinické, tak i klinické populace. Toto je však předmětem zkoumání navazující víceoborové spolupráce.

Diagnostický nástroj bude rovněž využit jako modul v neurorehabilitačním softwaru Eddie, kde bude nasazen při diagnostice pacientů se získaným poškozením mozku. Na základě výsledků

testů, které jsou realizovány ve vytvořené webové aplikaci, pak bude možno administrovat relevantní neurorehabilitační úlohy individuálně dle potřeb každého pacienta.

Při implementaci aplikace jsem se setkal s několika překážkami, které se mi podařilo vyřešit postupem, který byl pro mně zcela nový. Příkladem zde může být problém, jak zabránit manipulace výsledků testu uživateli skrze developerskou konzoli webového prohlížeče či jak zajistit bezpečnou autentizaci skrze cookie soubory a nahradit relační proměnné dočasnými daty v databázi.

Jednu ze složitých věcí představovala taktéž komunikace a porozumění požadavků na aplikaci kladených lidmi, jejichž hlavní profese nespadá pod informační technologie. Uvědomil jsem si, že každý člověk pohlíží na danou problematiku zcela z jiného úhlu pohledu, a není jednoduché se vždy vzájemně pochopit.

Další vývoj práce bude probíhat zejména na diagnostickém nástroji. Již v tuto dobu se začíná objevovat poptávka po některých funkcích, o které by mohla být v budoucnu aplikace rozšířena. Příkladem může být umožnění správci vytvářet vlastní skupiny uživatelů. Takto by pak mohlo v aplikaci probíhat měření a následné nezávislé hodnocení různých uživatelských skupin, které se od sebe značně liší, například lidé zdraví a lidé s poškozením mozku.

Literatura

- [1] MATLIN, Margaret W. *Cognition*. 6th ed., New York: J. Wiley & Sons, 2005.
- [2] PLHÁKOVÁ, A. *Učebnice obecné psychologie*. Praha: Academia, 2003.
- [3] Processing Speed - Cognitive Skill. *Brain Training, Brain Games, Memory Games, and Brain Fitness with CogniFit* [online]. 2019. [cit. 28.03.2019]. Dostupné z: <https://www.cognifit.com/science/cognitive-skills/processing-speed>
- [4] Working Memory and Processing Speed. *Specific Learning Difficulties Association of NSWSPELD NSW* [online]. [cit. 28.03.2019]. Dostupné z: <https://speldnsw.org.au/conferences-enews-more-locked/attachment/working-memory-and-processing-speed/>
- [5] KAPLAN, Robert M., SACCUZZO, Dennis P. *Psychological Testing: Principles, Applications, and Issues*. 7th ed., Belmont, CA: Wadsworth Cengage Learning, 2009.
- [6] URBÁNEK, T., DENGLEROVÁ, D., ŠIRŮČEK, J. *Psychometrika: Měření v psychologii*. Praha: Portál, 2011.
- [7] URBINA, S. *Essentials of Psychological Testing*. Hoboken, NJ: Wiley, 2004.
- [8] Rorschach inkblot test. In: *wikipedia.org* [online]. [cit. 16.04.2019]. Dostupné z: https://en.wikipedia.org/wiki/Rorschach_test#/media/File:Rorschach_blot_01.jpg
- [9] Development of a cognitive function test using virtual reality technology: examination in healthy participants. *Taylor & Francis* [online]. [cit. 02.04.2019]. Dostupné z: <https://www.tandfonline.com/doi/abs/10.1080/13825585.2017.1351916>
- [10] Neurokognitivní laboratoř. *Neurokognitivní laboratoř* [online]. [cit. 03.04.2019]. Dostupné z: <http://nela.fno.cz/>
- [11] Development of a self-administered web-based test for longitudinal cognitive assessment. *Nature News* [online]. [cit. 03.04.2019]. Dostupné z: <https://www.nature.com/articles/srep19114>
- [12] Inspirováno vědou. *Mentem.cz. Trénink mozku a paměti efektivně* [online]. [cit. 19.04.2019]. Dostupné z: <https://www.mentem.cz/inspirovano-vedou/>
- [13] Web Application Definition. *The Tech Terms Computer Dictionary* [online]. 2019. [cit. 05.04.2019]. Dostupné z: https://techterms.com/definition/web_application
- [14] What is a Web Application? *Stackpath* [online]. 2019. [cit. 05.04.2019]. Dostupné z: <https://www.maxcdn.com/one/visual-glossary/web-application/>

- [15] What is Web application (Web app)? - Definition from WhatIs.com. *Software Quality information, news and tips - SearchSoftwareQuality* [online]. [cit. 05.04.2019]. Dostupné z: <https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>
- [16] An overview of HTTP - HTTP | MDN. [online]. 2005. [cit. 05.04.2019]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
- [17] Web Application Development - Guides, Resources & Best Practices. *Custom Web Application Development / Custom Database Software / San Diego* [online]. 1996. [cit. 07.04.2019]. Dostupné z: <https://www.comentum.com/guide-to-web-application-development.html>
- [18] Common client side web technologies. *Microsoft Docs* [online]. [cit. 07.04.2019]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/standard/modern-web-apps-azure-architecture/common-client-side-web-technologies>
- [19] HTML & CSS - W3C. *World Wide Web Consortium (W3C)* [online]. [cit. 08.04.2019]. Dostupné z: <https://www.w3.org/standards/webdesign/htmlcss>
- [20] CSS Variables. *Can I use... Support tables for HTML5, CSS3, etc* [online]. [cit. 08.04.2019]. Dostupné z: <https://caniuse.com/#feat=css-variables>
- [21] What Is PHP? *Learn to Code / Digital Skills are Job Skills / Skillcrush* [online]. 2012. [cit. 09.04.2019]. Dostupné z: <https://skillcrush.com/2012/04/11/php/>
- [22] What is WordPress and Why Should You Use It? *Learn to Code / Digital Skills are Job Skills / Skillcrush* [online]. 2012. [cit. 09.04.2019]. Dostupné z: <https://skillcrush.com/2018/10/05/what-is-wordpress/>
- [23] Introduction to Node.js. *Node.js* [online]. [cit. 09.04.2019]. Dostupné z: <https://nodejs.dev/>
- [24] Introduction to Java Web development - Tutorial. *Eclipse, Android and Java training and support* [online]. [cit. 09.04.2019]. Dostupné z: <https://www.vogella.com/tutorials/JavaWebTerminology/article.html>
- [25] What is ASP.NET? - Definition from Techopedia. *Techopedia - Where Information Technology and Business Meet* [online]. 2019. [cit. 09.04.2019]. Dostupné z: <https://www.techopedia.com/definition/3213/asp-net>
- [26] What is ASP.NET? *.NET* [online]. [cit. 09.04.2019]. Dostupné z: <https://dotnet.microsoft.com/learn/web/what-is-aspnet>
- [27] TĚŠITELOVÁ, V., POLICAR, R., et al. *Jak implementovat Nařízení evropského parlamentu a rady (EU) 2016/679*. Praha: Ministerstvo zdravotnictví ČR, 2018.
- [28] These Major U.S. News Sites Are Blocked in the EU. *Fortune* [online]. [cit. 10.04.2019]. Dostupné z: <http://fortune.com/2018/08/09/news-sites-blocked-gdpr/>

- [29] Pseudonymization, Anonymization and the GDPR. *TermsFeed* [online]. 2019. [cit. 11.04.2019]. Dostupné z: <https://www.termsfeed.com/blog/gdpr-pseudonymization-anonymization/>
- [30] What is Data Encryption? *Kaspersky Lab* [online]. [cit. 10.04.2019]. Dostupné z: <https://www.kaspersky.com/resource-center/definitions/encryption>
- [31] What Is HTTPS, and Why Should I Care? *How-To Geek - We Explain Technology* [online]. 2019. [cit. 10.04.2019]. Dostupné z: <https://www.howtogeek.com/181767/htg-explains-what-is-https-and-why-should-i-care/>
- [32] Symmetric Key Encryption - why, where and how it's used in banking. *Cryptomathic - Security Solutions* [online]. 1986. [cit. 10.04.2019]. Dostupné z: <https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking>
- [33] Asymmetric Encryption explained with the help of chocolate boxes *Hackernoon* [online]. [cit. 10.04.2019]. Dostupné z: <https://hackernoon.com/asymmetric-encryption-explained-using-chocolate-boxes-5a329ea6813e>
- [34] What Is Hashing? Under The Hood Of Blockchain - Blockgeeks. *Blockchain Training, Education and Courses - Blockgeeks* [online]. 2019. [cit. 10.04.2019]. Dostupné z: <https://blockgeeks.com/guides/what-is-hashing/>
- [35] Hashing Algorithms | *Jscrambler Blog* [online]. [cit. 10.04.2019]. Dostupné z: <https://blog.jscrambler.com/hashing-algorithms/>
- [36] Guide to CSRF (Cross-Site Request Forgery) | *Veracode* [online]. 2019. [cit. 10.04.2019]. Dostupné z: <https://www.veracode.com/security/csrf>
- [37] SQL Injection Cheat Sheet & Tutorial | *Veracode* [online]. 2019. [cit. 10.04.2019]. Dostupné z: <https://www.veracode.com/security/sql-injection>
- [38] Cross-Site Scripting (XSS) Cheat Sheet | *Veracode* [online]. 2019. [cit. 10.04.2019]. Dostupné z: <https://www.veracode.com/security/xss>
- [39] What is Data Backup? - Definition from Techopedia. *Techopedia - Where Information Technology and Business Meet* [online]. 2019. [cit. 11.04.2019]. Dostupné z: <https://www.techopedia.com/definition/23338/data-backup>
- [40] What is backup? - Definition from WhatIs.com. *Data Backup information, news and tips - SearchDataBackup* [online]. [cit. 11.04.2019]. Dostupné z: <https://searchdatabackup.techtarget.com/definition/backup>

- [41] What is Cold Backup? - Definition from Techopedia. *Techopedia - Where Information Technology and Business Meet* [online]. 2019. [cit. 11.04.2019]. Dostupné z: <https://www.techopedia.com/definition/13589/cold-backup>
- [42] Is RAID a Good Backup Substitute? *BackupAssist* [online]. 2017. [cit. 11.04.2019]. Dostupné z: <https://www.backupassist.com/blog/support/raid-or-backup-is-raid-a-good-backup-substitute/>
- [43] PREISS, M., BARTOŠ, A., ČERMÁKOVÁ, R. *Neuropsychologická baterie Psychiatrického centra Praha*. 3. vydání. Praha: Psychiatrické centrum, 2013.
- [44] Tic Tac Toe - Creating Unbeatable AI. *Towards Data Science* [online]. [cit. 13.04.2019]. Dostupné z: <https://towardsdatascience.com/tic-tac-toe-creating-unbeatable-ai-with-minimax-algorithm-8af9e52c1e7d>
- [45] McGrew, K.S., 2009. CHC theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research.
- [46] MALANÍKOVÁ, Eva. SW aplikace pro ověření kognitivních funkcí. Ostrava, 2019. Bakalářská práce. VŠB - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky.
- [47] Pearson Correlation Coefficient: Formula, Example & Significance - Video & Lesson Transcript. *Study.com*. [online]. 2003. [cit. 21.04.2019]. Dostupné z: <https://study.com/academy/lesson/pearson-correlation-coefficient-formula-example-significance.html>

A Ukázka vyhodnocení testu

Výsledky testů

Výsledky vycházejí z normy, která se počítá z výsledků testů lidí, kteří spadají do stejných kategorií jako vy. Pokud je vaše norma počítána z malého počtu lidí, výsledky jsou nepřesné. Norma se průběžně aktualizuje s počtem lidí, kteří tento test vykonali.

Vyberte verzi testu, pro kterou chcete zobrazit výsledky.

✓ Test - verze A

Kvantitativní znalosti

Je to schopnost chápat numerické pojmy a vztahy a používat numerické symboly.

V této oblasti jste v normě.

Čtení a psaní

Zahrnují základní schopnosti číst a psát.

V této oblasti jste v normě.

Schopnost porozumět

Zahrnuje šířku a hloubku získaných vědomostí člověka, schopnost tyto vědomosti předat jiným lidem a schopnost použít tyto vědomosti při uvažování.

V této oblasti jste v normě.

Přízpůsobivé myšlení

Jedná se o schopnost zdůvodňovat, vytvářet představy a řešit problémy užitím neznámých informací a nových postupů.

V této oblasti vynikáte.

Krátkodobá paměť

Schopnost uchopit a udržet informaci a za krátký moment ji použít.

V této oblasti vynikáte.

Dlouhodobá paměť a vybavování

Jedná se o schopnost udržet informaci a plynule si ji vybavit později v procesu myšlení.

V této oblasti vynikáte.

Zrakové procesy

Je to schopnost vnímání, analýzy, syntézy a myšlení s obrazovými vzory včetně schopnosti uchovat tyto obrazové informace a později si je vybavit.

V této oblasti jste v normě.

Sluchové procesy

Analýza, syntéza a rozlišení zvukových podnětů včetně schopnosti zpracovat a rozlišit zvuky řeči i v případě zkreslení těchto zvukových podnětů.

V této oblasti vynikáte.

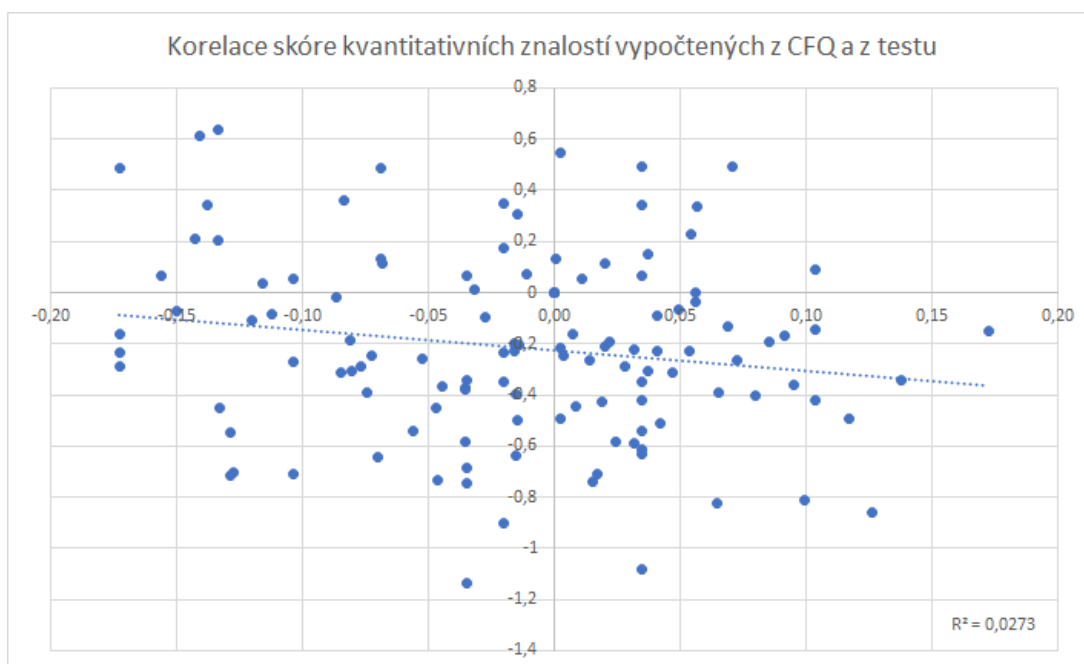
Rychlost zpracování informací

Schopnost provádět činnosti využívající kognitivní procesy při časovém tlaku.

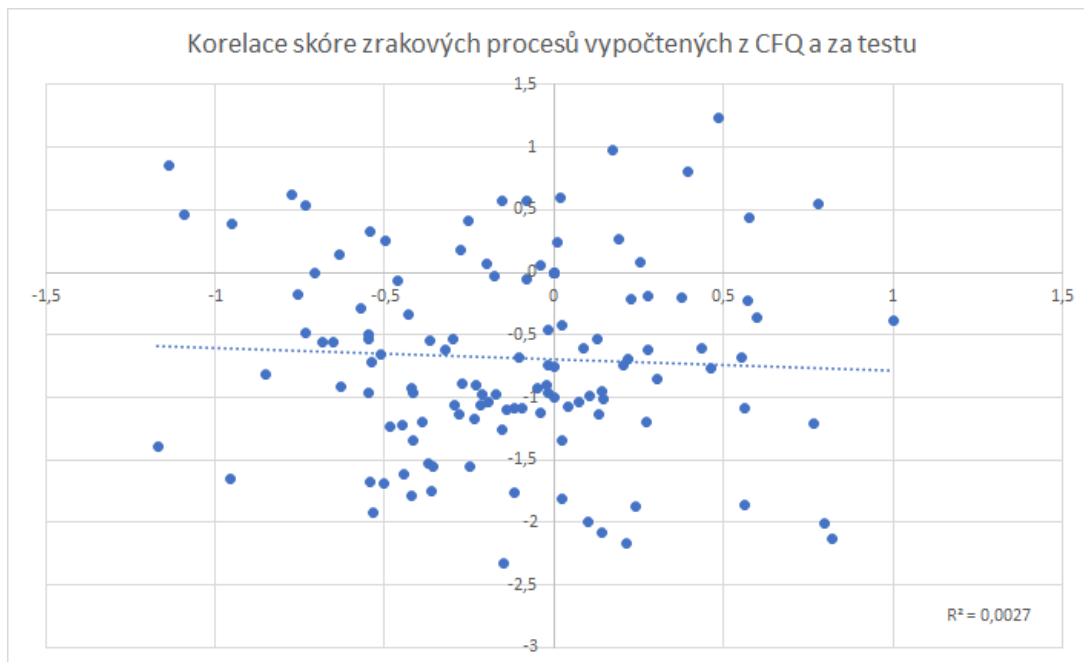
V této oblasti vynikáte.

Obrázek 20: Vyhodnocení testu

B Grafy korelací dvou měřených oblastí CHC modelu

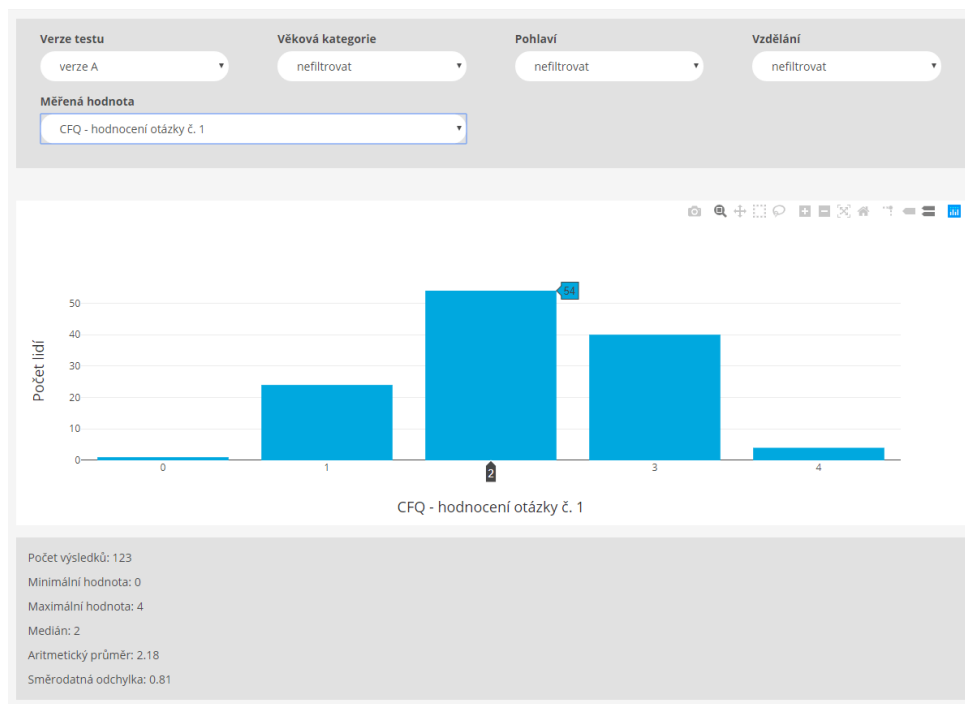


Obrázek 21: Graf korelace kvantitativních schopností

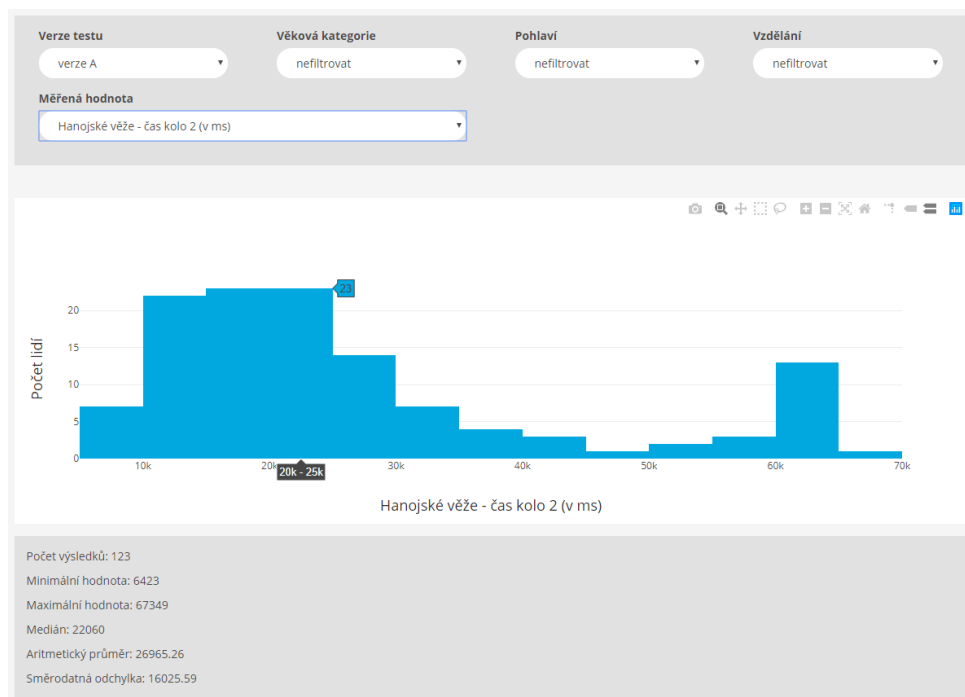


Obrázek 22: Graf korelace zrakových procesů

C Grafy zobrazené ve webové aplikaci



Obrázek 23: Sloupcový graf distribuce měřených položek v aplikaci



Obrázek 24: Histogram distribuce měřených položek v aplikaci

D Seznam elektronických příloh

Následující přílohy jsou k dispozici v IS EDISON.

- Aplikace
- Uživatelská příručka
- Programátorská dokumentace